

## Contents

Network Devices .....	4
Interfaces & Cables .....	6
<b>OSI &amp; TCP/IP .....</b>	<b>13</b>
<b>Intro to CLI .....</b>	<b>19</b>
<b>Ethernet LAN Switching .....</b>	<b>29</b>
Ethernet LAN Switching (Part 1) .....	29
Ethernet LAN Switching (Part 2) .....	34
<b>IPv4 Addressing .....</b>	<b>42</b>
IPv4 Addressing (Part 1).....	42
IPv4 Addressing (Part 2).....	47
<b>Switch Interfaces .....</b>	<b>52</b>
<b>IPv4 Header .....</b>	<b>59</b>
<b>Routing Fundamentals .....</b>	<b>65</b>
Routing Fundamentals.....	65
Static Routing.....	70
Life of a Packet .....	76
<b>Subnetting.....</b>	<b>78</b>
<b>VLAN .....</b>	<b>88</b>
VLAN Intro.....	88
Trunk Ports .....	92
SVI.....	103
DTP & VTP .....	107
<b>Spanning Tree Protocol (STP) .....</b>	<b>117</b>
STP (Part 1).....	117
STP (Part 2).....	125
STP Additional Features .....	133
Rapid Spanning Tree .....	141
<b>EtherChannel.....</b>	<b>147</b>
<b>Dynamic Routing.....</b>	<b>156</b>
Intro to Dynamic Routing.....	156

RIP & EIGRP .....	164
<b>OSPF .....</b>	<b>169</b>
OSPF (Part 1) .....	169
OSPF (Part 2) .....	177
<b>FHRP .....</b>	<b>201</b>
<b>TCP/UDP .....</b>	<b>211</b>
<b>IPv6 Addressing .....</b>	<b>218</b>
IPv6 (Part 1) .....	218
IPv6 (Part 2) .....	222
<b>Access Control Lists (ACLs) .....</b>	<b>239</b>
Standard ACL .....	239
Extended ACLs .....	247
<b>Layer 2 Discovery Protocols (CDP &amp; LLDP) .....</b>	<b>255</b>
<b>Network Time Protocol (NTP) .....</b>	<b>265</b>
<b>Domain Name System (DNS).....</b>	<b>275</b>
<b>Dynamic Host Configuration Protocol (DHCP) .....</b>	<b>282</b>
<b>SNMP (Simple Network Management Protocol) .....</b>	<b>292</b>
<b>Syslog.....</b>	<b>298</b>
<b>SSH &amp; Telnet .....</b>	<b>303</b>
<b>FTP/TFTP.....</b>	<b>309</b>
<b>Network Address Translation (NAT) .....</b>	<b>318</b>
Static NAT .....	318
Dynamic NAT / PAT.....	322
<b>Quality of Service (QoS).....</b>	<b>329</b>
Intro to QoS.....	329
QoS (Part 2).....	335
<b>Security Fundamentals .....</b>	<b>344</b>
<b>Port Security .....</b>	<b>352</b>
<b>DHCP Snooping .....</b>	<b>361</b>
<b>Dynamic ARP Inspection .....</b>	<b>368</b>
<b>Network Architectures .....</b>	<b>377</b>

LAN Architectures .....	377
WAN Architectures .....	383
<b>Virtualization &amp; Cloud.....</b>	<b>393</b>
<b>Virtual Routing&amp; Forwarding (VRF).....</b>	<b>404</b>
<b>Wireless .....</b>	<b>407</b>
Wireless Fundamentals .....	407
Wireless Architecture .....	419
Wireless Security.....	429
<b>Network Automation .....</b>	<b>436</b>
Network Automation.....	436
AI & Machine Learning .....	441
JSON, XML, YAML .....	445
REST APIs.....	448
REST API Authentication.....	453
Software Defined Networking .....	457
Ansible, Puppet, Chef .....	463
Terraform .....	468

# Network Devices

## **Network**

- A computer network is a digital telecommunications network which allows nodes to share resources

## **Client**

- A device that accesses a service made available by a server

## **Server**

- A device that provide functions or services to clients
- Can be as simple as a smartphone or pc

## **Switch**



Catalyst 9200



Catalyst 3650

- Have many network interfaces/port for end hosts to connect (usually 24+)
- Provide connectivity to hosts within the same LAN (Local Area Network)
- Do not provide connectivity btw LAN or the Internet

## **Router**



ISR 1000



ISR 900



ISR 4000

- Fewer network interfaces than switches
- Used to provide connectivity btw LANs

- Therefore used to send data over the Internet
- Can provide some basic security features

## Firewalls

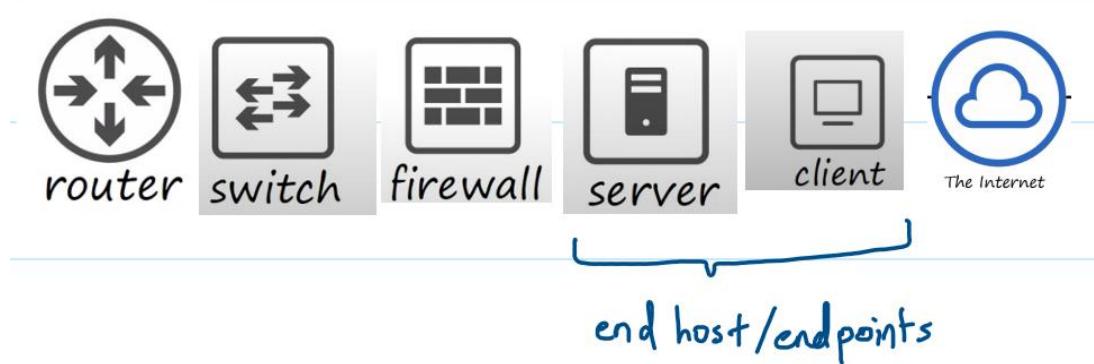


ASA5500-X



Firepower 2100

- Monitor and control network traffic based on configured rules
- Can be placed 'inside' or 'outside' the network
- Are known as 'Next-Generation Firewalls' when they include more modern and advanced filtering capabilities
- The 2 firewalls above are next-gen
- Types of firewalls:
  - **Network firewalls**
    - Hardware devices that filter traffic between networks
  - **Host-based firewalls**
    - Software applications that filter traffic entering and exiting a host machine, like a PC



## Interfaces & Cables

### RJ45

- Registered Jack
- Used at the end of a copper ethernet cable



### Ethernet

- A collection of network protocols/standards
- For the lesson, we will focus on type of cabling for ethernet

### Network Protocols

- Allow devices to communicate with each other without issues

### Bits & Bytes

- 1 byte = 8 bit
- Speed measured in bits/s
- Data on drives measured in bytes

### Ethernet Standards

- Defined in the IEEE 802.3 standard
- IEEE: Institute of Electrical and Electronics Engineers
- Ethernet Standards (Copper)

Speed	Common Name	IEEE Standard	Informal Name	Maximum Length
10 Mbps	Ethernet	802.3i	10BASE-T	100 m
100 Mbps	Fast Ethernet	802.3u	100BASE-T	100 m
1 Gbps	Gigabit Ethernet	802.3ab	1000BASE-T	100 m
10 Gbps	10 Gig Ethernet	802.3an	10GBASE-T	100 m

- BASE: refers to baseband signalling
- T: twisted pair

### UTP Cables

- Unshielded Twisted Pair
- The twist protects against Electromagnetic Interference



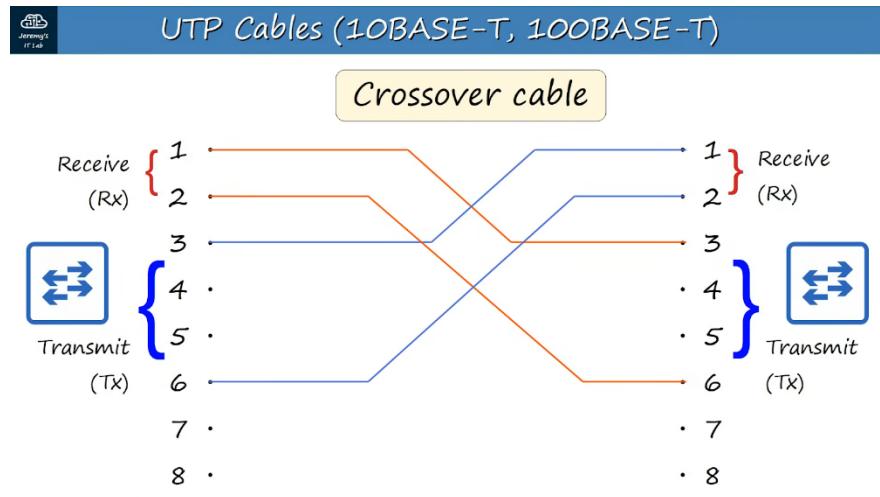
- Consists of 8 wires
  - 10/100BASE-T: 2 pairs (4 wires)
  - 1000/10GBASE-T: 4 pairs (8 wires)
- 10/100BASE-T
  - Straight-through cable
    - Pin 1 -> Pin 1

## Full-Duplex



- Crossover cable

- Pin 1 → Pin 3





## UTP Cables (10BASE-T, 100BASE-T)

Device Type	Transmit (Tx) Pins	Receive (Rx) Pins	
Router		1 and 2	3 and 6
Firewall		1 and 2	3 and 6
PC		1 and 2	3 and 6
Switch		3 and 6	1 and 2

- Auto MDI-X
  - New network devices has a feature called Auto MDI-X that allows the transmit and receive pins to be connected incorrectly and still work
- 1000/10GBASE-T

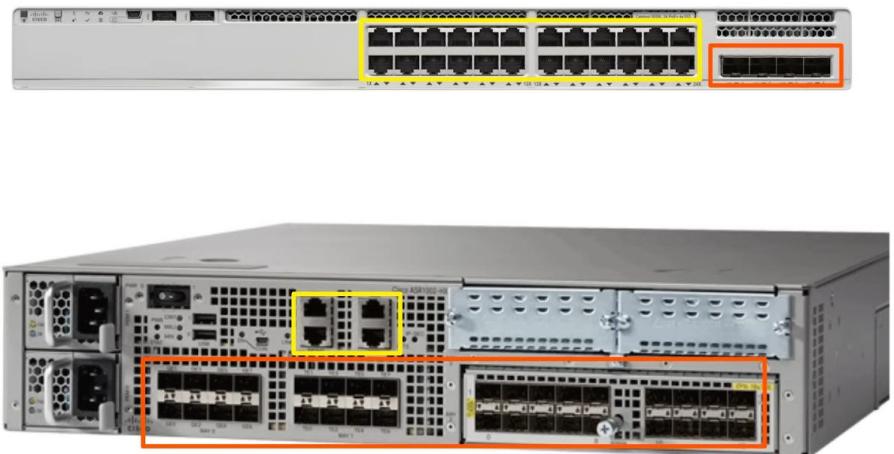


## UTP Cables (1000BASE-T, 10GBASE-T)

Each pair is bidirectional.



## Fibre-Optic Connections

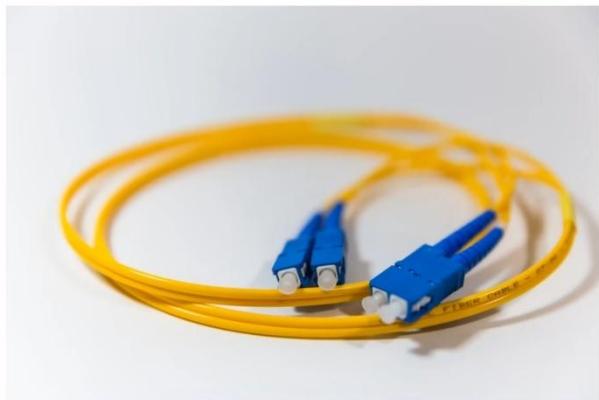


## SFP Transceiver

(Small Form-Factor Pluggable)

- Plugged into the fibre optic connection
- They are connected to fibre optic cables

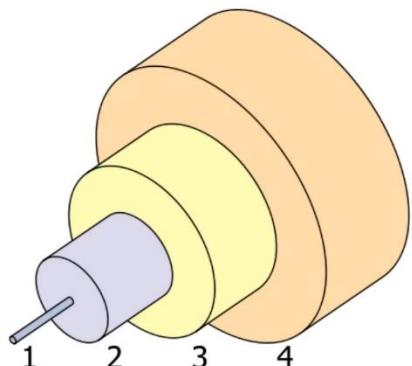
Fibre Optic Cables



\



## Fiber-Optic Connections

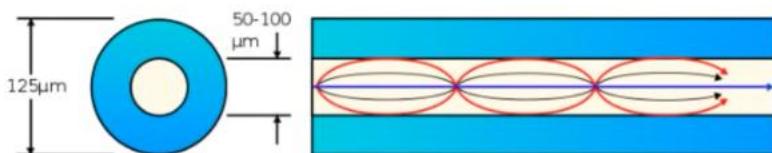
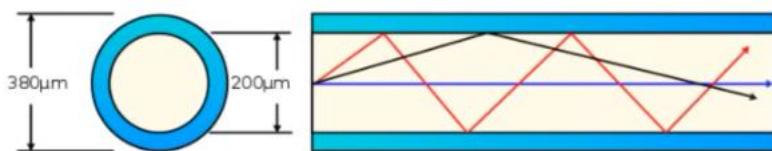


- 1: the fiberglass core itself
- 2: cladding that reflects light
- 3: a protective buffer
- 4: the outer jacket of the cable

Original by Bob Mellish, SVG derivative by BenChil  
([https://commons.wikimedia.org/wikifile/Singlemode\\_fibre\\_structure.svg](https://commons.wikimedia.org/wikifile/Singlemode_fibre_structure.svg)), „Singlemode fibre structure“, <https://creativecommons.org/licenses/by-sa/3.0/legalcode>

- They are different types of fibre cables
  - Single-mode
  - Multi-mode

- Multi Mode



- Core diameter is wider than single-mode fibre
- Allow multiple angles (modes) of light waves to enter the fibreglass core
- Allow longer cables than UTP, but shorter than single-mode
- Cheaper than single-mode (due to cheaper LED-based SFP transmitter)
- Single-Mode



- Core diameter is narrower than multi-mode fibre
- Light enters from a single angle (mode) from a laser based transmitter
- Allow longer cables than both UTP and multi-mode fibre
- More expensive than multi-mode fibre (due to more expensive laser-based SFP transmitter)



## Fiber-Optic Cable Standards

Informal Name	IEEE Standard	Speed	Cable Type	Maximum Length
1000BASE-LX	802.3z	1 Gbps	Multimode or Single-Mode	550 m (MM) 5 km (SM)
10GBASE-SR	802.3ae	10 Gbps	Multimode	400 m
10GBASE-LR	802.3ae	10 Gbps	Single-Mode	10 km
10GBASE-ER	802.3ae	10 Gbps	Single-Mode	30 km



## UTP vs Fiber-Optic Cabling

### UTP

- Lower cost than fiber-optic.
- Shorter maximum distance than fiber-optic (~100m).
- Can be vulnerable to EMI (Electromagnetic Interference).
- RJ45 ports used with UTP are cheaper than SFP ports.
- Emit (leak) a faint signal outside of the cable, which can be copied (=security risk)

### Fiber-Optic

- Higher cost than UTP.
- Longer maximum distance than UTP.
- No vulnerability to EMI.
- SFP ports are more expensive than RJ45 ports (single-mode is more expensive than multimode).
- Does not emit any signal outside of the cable (=no security risk).

## OSI & TCP/IP

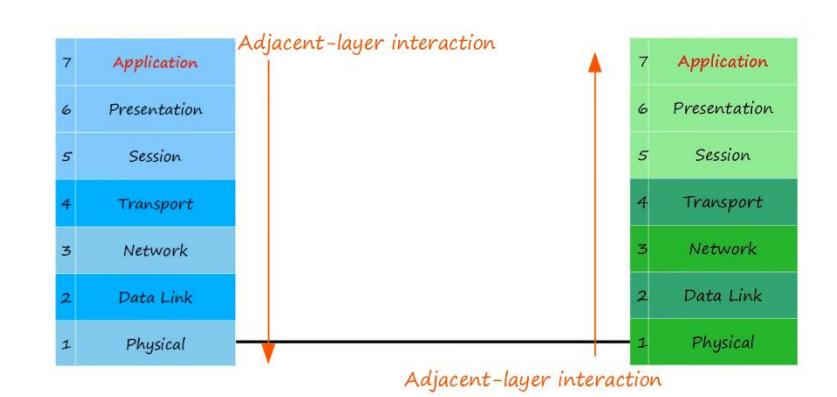
### What is a networking model?

- Networking models categorize and provide structure for networking protocols and standards

- Protocols
  - A set of logical rules (non-physical) defining how a network device and software should work

## OSI Model

- Open Systems Interconnection model
- A conceptual model that categorizes and standardizes the different functions in a network
- Divided into 7 layers
- The layers work together to make the network work
- Layer 7: Application
  - Layer closest to end user
  - Interacts with software applications (e.g. web browser - chrome)
  - HTTP and HTTPS are Layer 7 protocols
  - Doesn't include the applications, but rather the protocols that interact with the app like HTTP
  - Functions include
    - Identifying communication partners
    - Synchronising communication



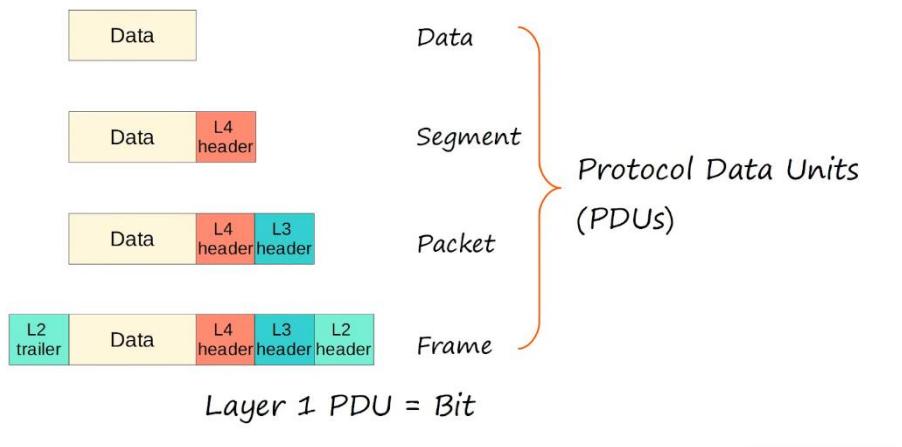
- Same-layer interaction in the application layer allows the layer 7 functions
- Layer 6: Presentation

- Data in the presentation layer is in 'application format'
  - It needs to be translated to a different format to be sent over the network
  - The presentation layer's job is to translate btw application and network formats
  - Eg. Encryption of data as it is sent, and decryption as it is received
  - Translates btw different application-layer formats
  - Basically - translates data into the appropriate format
- Layer 5: Session
  - Control dialogues (sessions) btw communicating hosts
  - Establishes, manages, and terminates connections btw the local application (e.g. your web browser) and the remote application (e.g. YouTube)
  - YouTube services are used by many people and there is a need to manage the sessions, which is the purpose of the Session Layer
- Network engineers don't usually work with top 3 (upper) layers
- Application developers work with the upper layers to connect their applications over networks
- Layer 4: Transport
  - Segments and reassembles data for communications btw end hosts
  - Break large pieces of data into smaller segments which can be more easily sent over the network and are less likely to cause transmission problems if errors occur
    - e.g. If want to watch video but there is error, cannot watch at all. But if video data divided into small units, still can watch some
  - Provides host-to-host communication
- Layer 3: Network
  - Provides connectivity btw end hosts on different networks (e.g. outside of LAN)

- Provides logical addressing (IP addresses)
  - Provides path selection btw source and destination
    - There are many path the data can take, layer 3 selects the best path
  - Routers operate at layer 3
- Layer 2: Data Link
  - Provides node-to-node connectivity and data transfer (e.g. PC to switch, switch to router, router to router)
  - Defines how data is formatted for transmission over a physical medium (e.g. copper UTP cables)
  - Detects and (possibly) corrects Physical Layer errors
  - Uses Layer 2 addressing, separate from Layer 3 addressing
  - Switches operate at Layer 2
- Layer 1: Physical
  - Defines the physical characteristics of the medium used to transfer data btw devices
    - E.g. Voltage levels, max transmission distances, physical connectors, cable specifications, etc
  - Digital bits are converted to electrical (for wired connections) or radio (for wireless connections) signals
  - All the info in day 2 is related to Layer 1

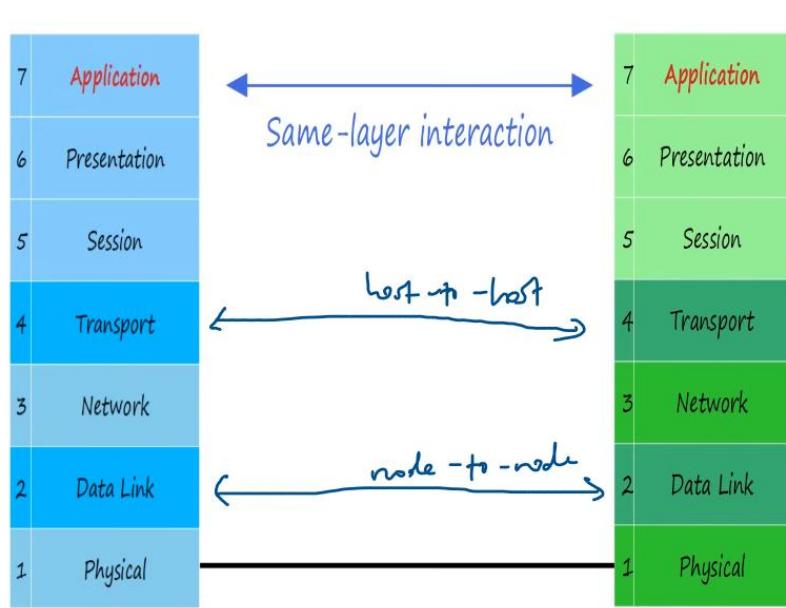


## OSI Model – PDUs



## Acronyms

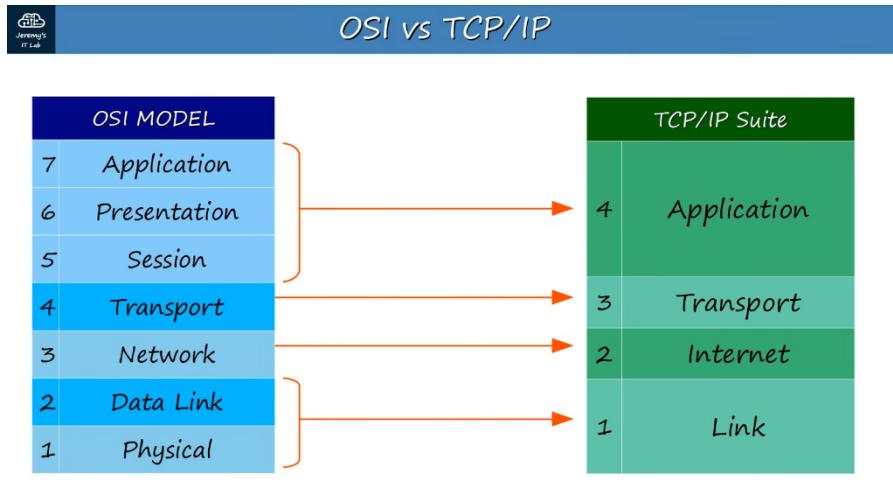
- All People Seem To Need Data Processing
- Please Do Not Teach Students Pointless Acronyms



## TCP/IP suite

- Conceptual model and set of communications protocols used in the Internet and other networks
- Known as TCP/IP because those are 2 of the foundational protocols in the suite
- Similar to OSI but lesser layers

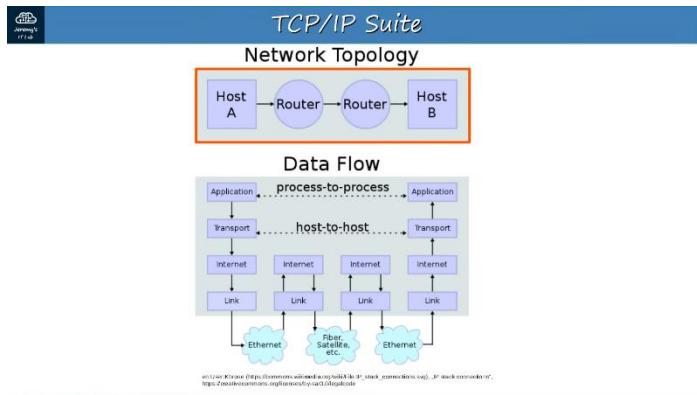
- Used in modern networks
- OSI model still influences how network engineers think and talk about networks



- When talking about layers, refer to the OSI
  - E.g. there is a layer 4 problem - refers to transport layer

### Different naming conventions for the TCP/IP suites

RFC 1122[5], Internet STD 3 (1989)	Cisco Academy[31]	Kurose [32] Forouzan [33]	Comer,[34] Kozirok[35]	Stalling[36]	Tanenbaum[37]	Arpanet Reference Model (RFC 871[2])	OSI model
Four layers	Four layers	Five layers	Four+one layers	Five layers	Five layers	Three layers	Seven layers
"Internet model"	"Internet model"	"Five-layer Internet model" or "TCP/IP protocol suite"	"TCP/IP 5-layer reference model"	"TCP/IP model"	"TCP/IP 5-layer reference model"	"Arpanet reference model"	OSI model
Application	Application	Application	Application	Application	Application	Application/Process	Application
Transport	Transport	Transport	Transport	Host-to-host or transport	Transport	Host-to-host	Transport
Internet	Internetwork	Network	Internet	Internet	Internet	Host-to-host	Network
Link	Network interface	Data link	Data link (Network interface)	Network access	Data link	Network interface	Data link
		Physical	(Hardware)	Physical	Physical	Physical	Physical



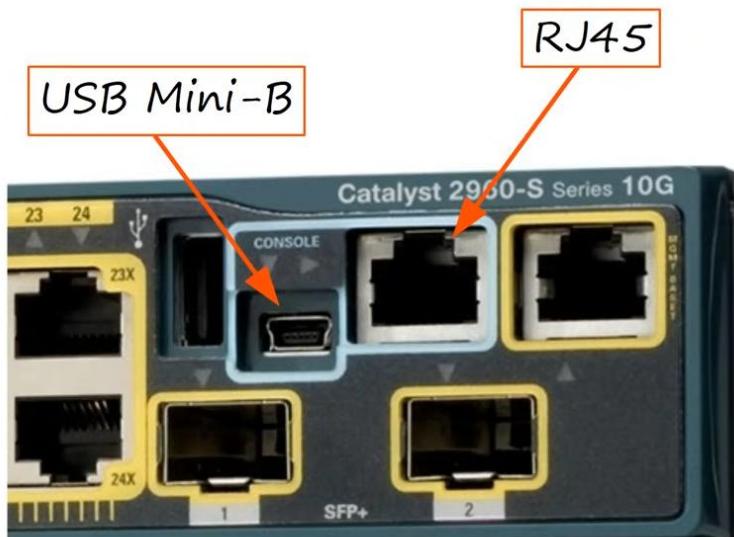
- De-encapsulate till the internet layer only as they only need the IP address to know where to send to

## Intro to CLI

What is a CLI?

- Command Line Interface
- Used to configure Cisco devices
- GUI: Graphical User Interface

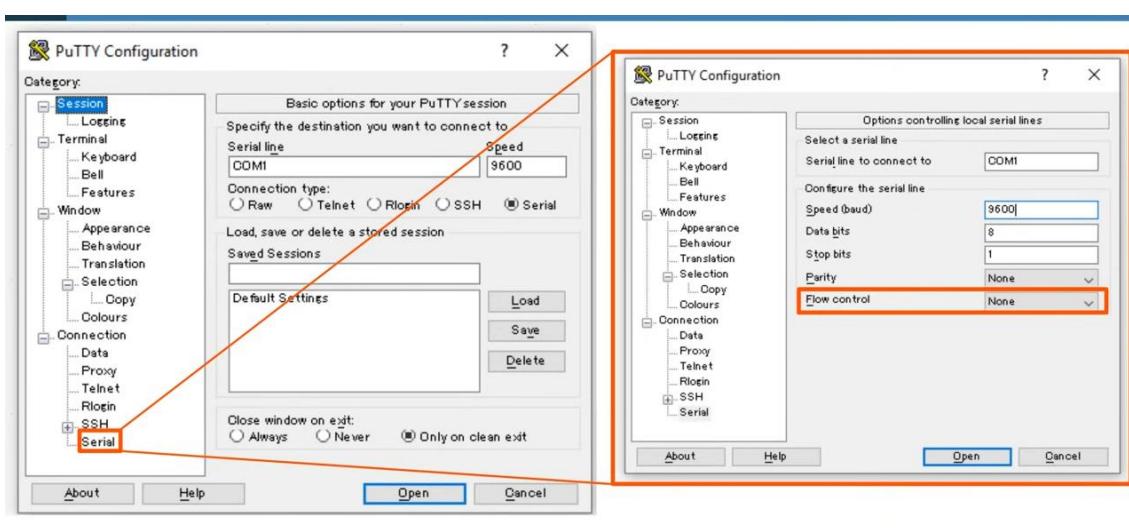
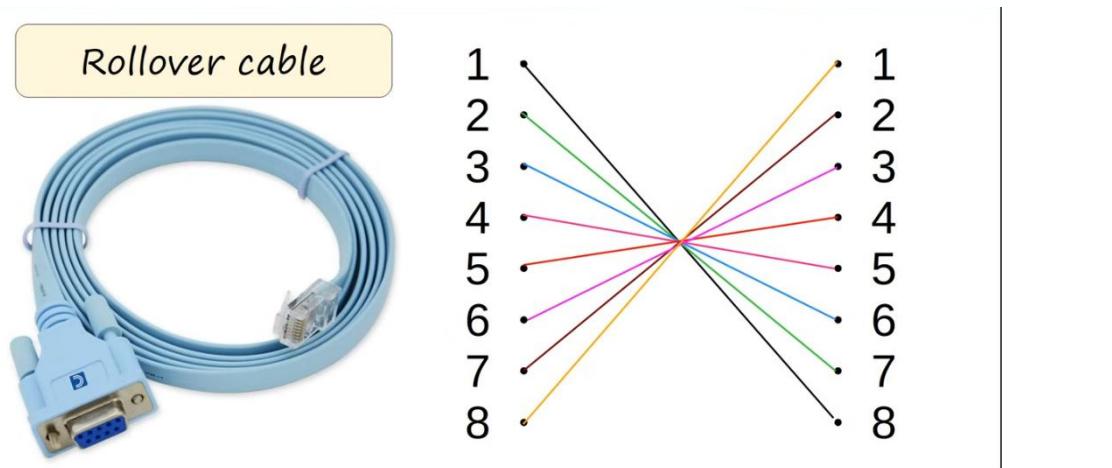
How to connect a Cisco device? (Console port)



- Can connect to either of them



- Will need to use a rollover cable



- Connect to CLI using PuTTY
- Settings (default settings)
  - Serial

- Speed (baud) : 9600
- Data bits: 8
- Stop bits: 1
- Parity: None
- Flow Control: None

## User EXEC Mode

```

Impressors, exporters, distributors and users are responsible for
compliance with U.S. and local laws before ordering this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com

Cisco CISCO2911/K9 (revision 1.0) with 491520K/32768K bytes of memory.
Processor board ID FTX152400KS
3 Gigabit Ethernet interfaces
DRAM configuration is 64 bits wide with parity disabled.
256K bytes of non-volatile configuration memory.
249856K bytes of ATA System CompactFlash 0 (Read/Write)

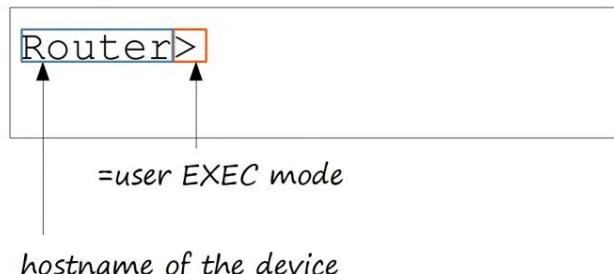
--- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: no

Press RETURN to get started!

Router>

```



- User EXEC mode is very limited
- Users can look at some things, but can't make any changes to configuration
- Also called 'user mode'

## Privileged EXEC Mode

```

To comply with U.S. and local laws, return this product immediately.
A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com

Cisco CISCO2911/K9 (revision 1.0) with 491520K/32768K bytes of memory.
Processor board ID FTX152400KS
3 Gigabit Ethernet interfaces
DRAM configuration is 64 bits wide with parity disabled.
256K bytes of non-volatile configuration memory.
249856K bytes of ATA System CompactFlash 0 (Read/Write)

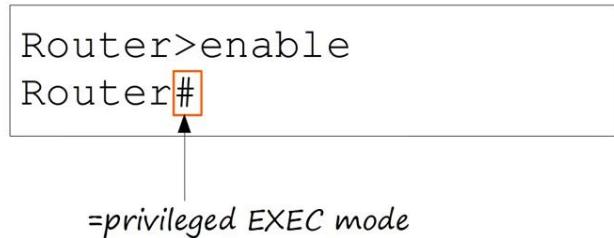
--- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: no

Press RETURN to get started!

Router>enable
Router#

```



- Provides complete access to view the device's configuration, restart the device, etc
- Cannot change the configuration, but can change the time on the device, save the configuration, etc

## User EXEC Mode

```
Router>?
Exec commands:
<1-99> Session number to resume
connect Open a terminal connection
disable Turn off privileged commands
disconnect Disconnect an existing network connection
enable Turn on privileged commands
exit Exit from the EXEC
logout Exit from the EXEC
ping Send echo messages
resume Resume an active network connection
show Show running system information
ssh Open a secure shell client connection
telnet Open a telnet connection
terminal Set terminal line parameters
traceroute Trace route to destination
Router>
```

## Privileged EXEC Mode

```
Router#?
Exec commands:
<1-99> Session number to resume
auto Exec level Automation
clear Reset functions
clock Manage the system clock
configure Enter configuration mode
connect Open a terminal connection
copy Copy from one file to another
debug Debugging functions (see also 'undebug')
delete Delete a file
dir List files on a filesystem
disable Turn off privileged commands
disconnect Disconnect an existing network connection
enable Turn on privileged commands
erase Erase a filesystem
exit Exit from the EXEC
logout Exit from the EXEC
mkdir Create new directory
more Display the contents of a file
no Disable debugging informations
ping Send echo messages
reload Halt and perform a cold restart
resume Resume an active network connection
rmdir Remove existing directory
send Send a message to other tty lines
setup Run the SETUP command facility
show Show running system information
ssh Open a secure shell client connection
telnet Open a telnet connection
terminal Set terminal line parameters
traceroute Trace route to destination
undebug Disable debugging functions (see also 'debug')
vlan Configure VLAN parameters
write Write running configuration to memory, network, or terminal
Router#
```

- Use 'en' or 'enable' to go to privileged mode

## Global Configuration Mode

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config) #
```

```
Router>enable
Router#con?
configure connect
Router#conf t?
terminal
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config) #
```

## Enable Password

```
Router(config)#enable password?  
password  
Router(config)#enable password ?  
7      Specifies a HIDDEN password will follow  
LINE   The UNENCRYPTED (cleartext) 'enable' password  
level  Set exec level password  
Router(config)#enable password CCNA ?  
<cr>  
Router(config)#enable password CCNA  
Router(config)#+
```

- Passwords are case sensitive

```
Router(config)#enable password CCNA  
Router(config)#exit  
Router#  
%SYS-5-CONFIG_I: Configured from console by console  
Router#exit
```

```
Router con0 is now available
```

```
Press RETURN to get started.
```

```
Router>enable  
Password:   
Router#
```

- The password does **not** display as you type it (for security purposes).

```
Router>enable  
Password:  
Password:  
Password:  
% Bad secrets  
  
Router>
```

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#enable password CCNA
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#exit
```

```
Router>enable
Password:
Router#
```

## running-config / startup-config

- There are 2 separate config files kept on the device at once
- Running-config
  - The current, active config file on the device
  - As you enter commands in the CLI, you edit the active config
- Startup-config
  - The config file that will be loaded upon restart of the device

```
Router#show running-config
Building configuration...

Current configuration : 714 bytes
!
version 15.1
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Router
!
!
!
enable password CCNA
!
```

```
Router#show startup-config
startup-config is not present
```

---

Saving the configuration

```
Router#write
Building configuration...
[OK]
Router#write memory
Building configuration...
[OK]
Router#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
Router#
```

Service password-encryption

```
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#service password-encryption

Router#show running-config
Building configuration...

Current configuration : 719 bytes
!
version 15.1
no service timestamps log datetime msec
no service timestamps debug datetime msec
service password-encryption
!
hostname Router
!
!
!
enable password 7 08026F6028
!
```

- Encrypted password can easily be decrypted by basic online tools

#### Enable secret

---

```
Router(config)#enable secret Cisco
Router(config)#do sh run
Building configuration...

Current configuration : 766 bytes
!
version 15.1
no service timestamps log datetime msec
no service timestamps debug datetime msec
service password-encryption
!
hostname Router
!
!
!
enable secret 5 $1$xEERr$YlCKLMcTYWwkF1Cndtll.
enable password 7 08026F6028
```

- 
- At the last 2 lines, both "enable secret" and "enable password" shown
  - It will automatically choose the "enable secret" and ignore "enable password"

- "enable secret" will always be encrypted, don't need to use "service password-encryption" command as previously used with "enable password"
- Should always use "enable secret"

## Cancelling Commands

```

Router(config)#no service password-encryption
Router(config)#do sh run
Building configuration...

Current configuration : 769 bytes
!
version 15.1
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Router
!
!
!
enable secret 5 $1$xEERr$Y1CKLMcTYWwkF1Cndtll.
enable password 7 08026F6028
!
```

- 
- Write a "no" in front of the command you want to cancel

## "service password-encryption"

- If enable "service password-encryption"
  - Current passwords will be encrypted
  - Future passwords will be encrypted

- "enable secret" will not be affected
- If disable
  - Current passwords remain encrypted and will not be decrypted
  - Future passwords will not be encrypted
  - "enable secret" will not be affected

## Modes Review

- Router> : user EXEC mode
- Router# : privileged EXEC mode
- Router(config)# : global configuration mode

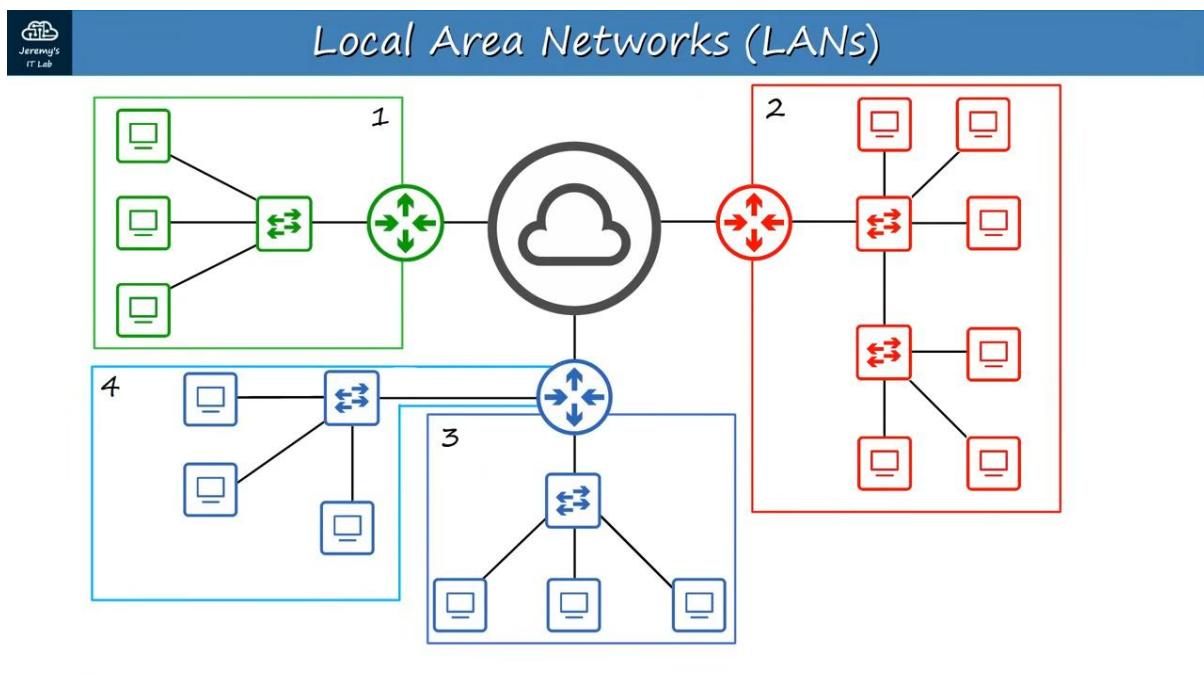
## Command Review

- Router>**enable**
  - Used to enter privileged EXEC mode
- Router#**configure terminal**
  - Used to enter global config mode
- Router(config)#**enable password** password
  - Configures a password to protect privileged EXEC mode
- Router(config)#**service password-encryption**
  - Encrypts the enable password (and other passwords)
- Router(config)#**enable secret** password
  - Configures a more secure, always encrypted enable password
- Router(config)#**run privileged-exec-level-command**
  - Executes a privileged-exec level command from global config mode
- Router(config)#**no** command

- Removes the command
- Router(config)#**show running-config**
  - Displays the current, active config file
- Router(config)#**show startup-config**
  - Displays the saved config file which will be loaded if the device is restarted
- Router(config)#**write**
  - Saves the configuration
- Router(config)#**write memory**
  - Saves the config
- Router(config)#**copy running-config startup-config**
  - Saves the config

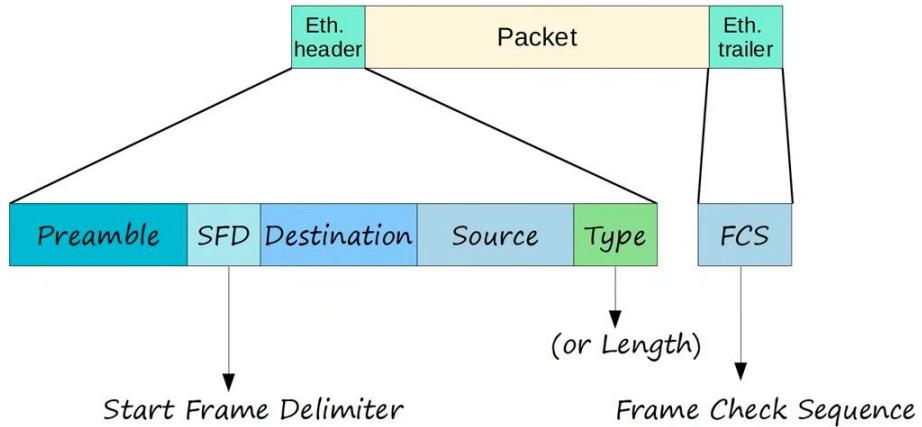
## Ethernet LAN Switching

### Ethernet LAN Switching (Part 1)



- When switches are connected to router but not connected to each other, they create separate LANs

## Ethernet Frame

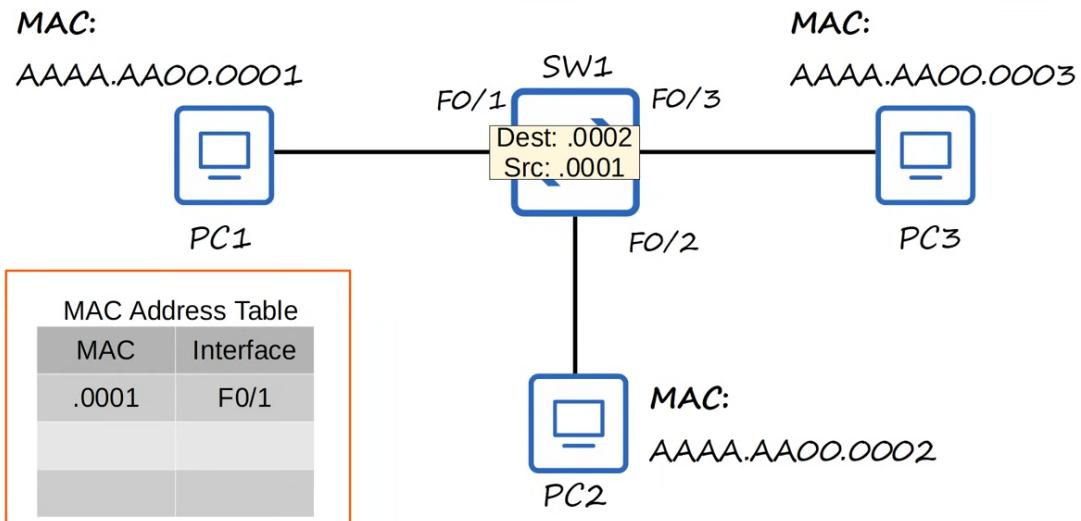


- Preamble
  - Length: 7 bytes = 56 bits
  - Alternating 1 and 0 ( $10101010 * 7$ )
  - Allow devices to synchronize their receiver clocks
- SFD
  - Start Frame Delimiter
  - Length: 1 byte = 8 bits
  - $10101011$
  - Indicates the end of the preamble and the beginning of the rest of the frame
- Destination & Source
  - Indicate the devices sending and receiving the frames
  - Consists of the destination and source MAC address
  - MAC = Media Access Control
  - 6 byte = 48 bit address of the physical device
- Type/Length
  - 2 bytes = 16 bits
  - Indicate type/length of encapsulated packet

- If the value of the field is 1500 or less, it indicates the LENGTH of the encapsulated packet in bytes
  - A value of 1536 or greater indicates the TYPE of the encapsulated packet (usually IPv4 or IPv6), and the length is determined via other methods
  - IPv4 = 0x0800 = 2048
  - IPv6 = 0x86DD = 34525
- FCS
  - Frame Check Sequence
  - 4 bytes = 32 bits
  - Detects corrupted data by running 'CRC' algorithm over the received data
  - CRC: Cyclic Redundancy Check
    - Cyclic refer to cyclic codes
    - Redundancy refers to the fact that the 4 bytes enlarges the frame w/o adding any new info
    - Check refers that is checks for error
- Total = 26 bytes (header + trailer)

## MAC Address

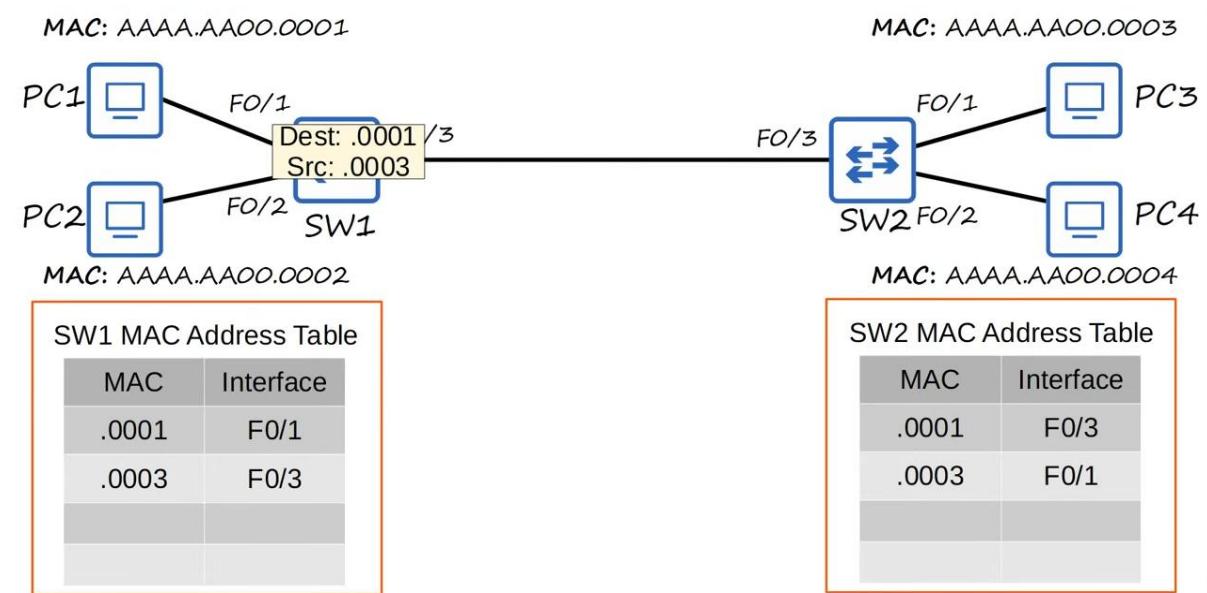
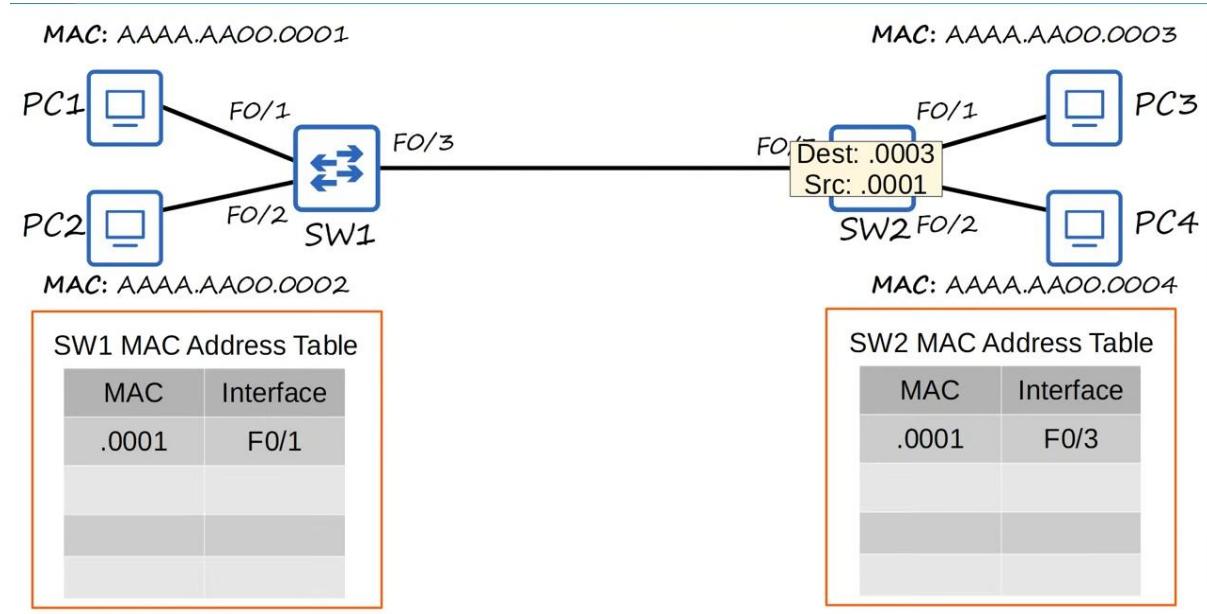
- 6 byte (48 bit) physical address assigned to the device when it is made
- A.K.A 'Burned in Address' (BIA)
- Globally unique
- First 3 bytes are the OUI (Organizationally Unique Identifier) which is assigned to the company making the device
- The last 3 bytes are unique to the device itself
- Written as 12 hexadecimal characters



- F0/1
  - Refers to the interface
  - 'F': Fast Ethernet - 100Mbps
- MAC Address Table
  - Dynamically learned MAC Address
  - Switch will know MAC address of source device only when receive frame from the source
- Unicast Frame: A frame destined for a single target
- Unknown unicast frame
  - When the destination is not in the MAC address table
  - Will need to FLOOD the frame
  - Sends the frame on all interface except the one it received from
  - PC3 ignores the frame
  - PC2 receives the packet and decapsulate the packet
  - If it does not send out any packet, the switch will not know the interface of the receiver
  - So, if PC1 sends a packet to PC2 again, will still need to flood the interface
- Known unicast frame
  - Destination in the MAC address table already

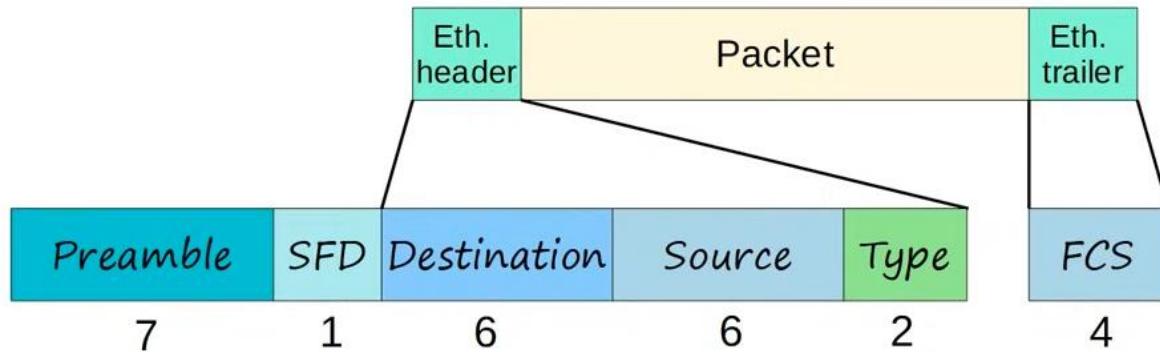
- FORWARD to the correct destination
- For Cisco devices, dynamic MAC address are removed from the MAC address table after 5 minutes of inactivity

### Example



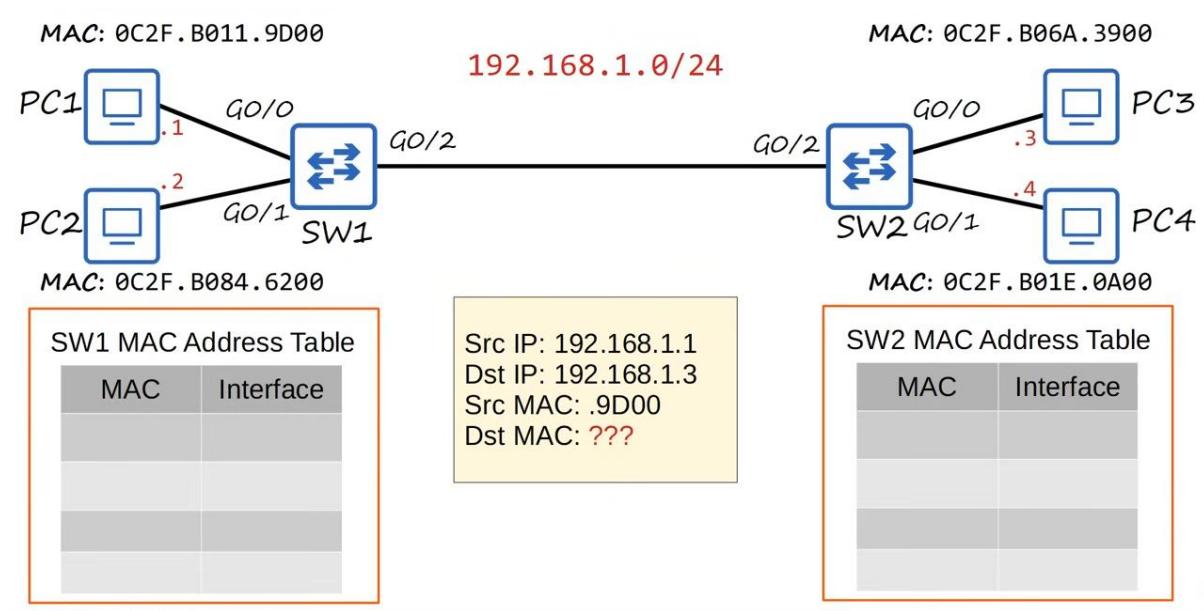
## Ethernet LAN Switching (Part 2)

### Ethernet Frame

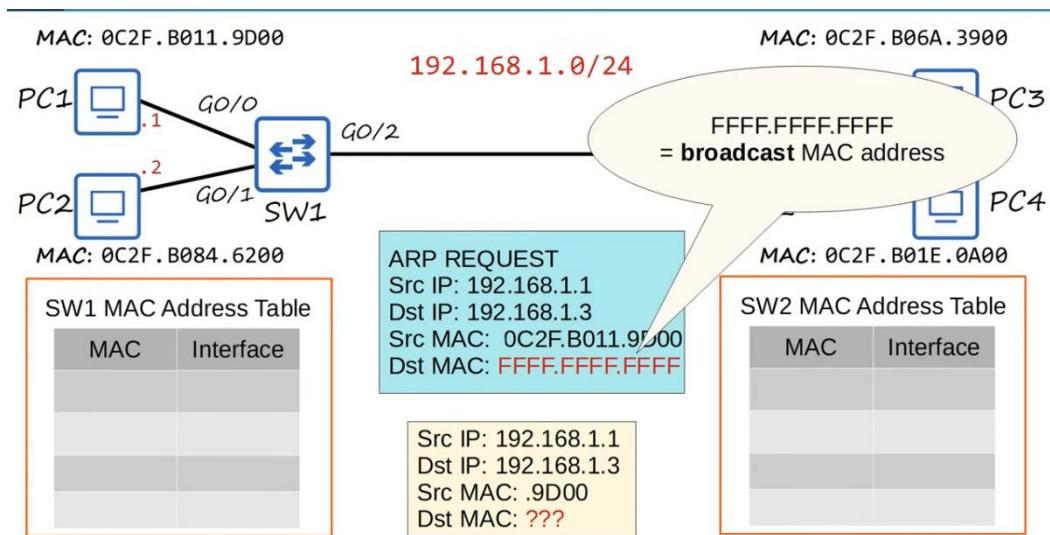


- The preamble + SFD usually not considered as part of the Ethernet header
  - Depends on how you define it
  - Still included in all Ethernet frames
- Size of Ethernet header + trailer = 18 bytes
- Minimum size for an Ethernet frame
  - 64 bytes
  - Header + Payload (packet) + Trailer
  - Minimum payload size =  $64 - 18 = 46$  bytes
  - If payload smaller than 46 bytes, padding bytes added (all 0's)

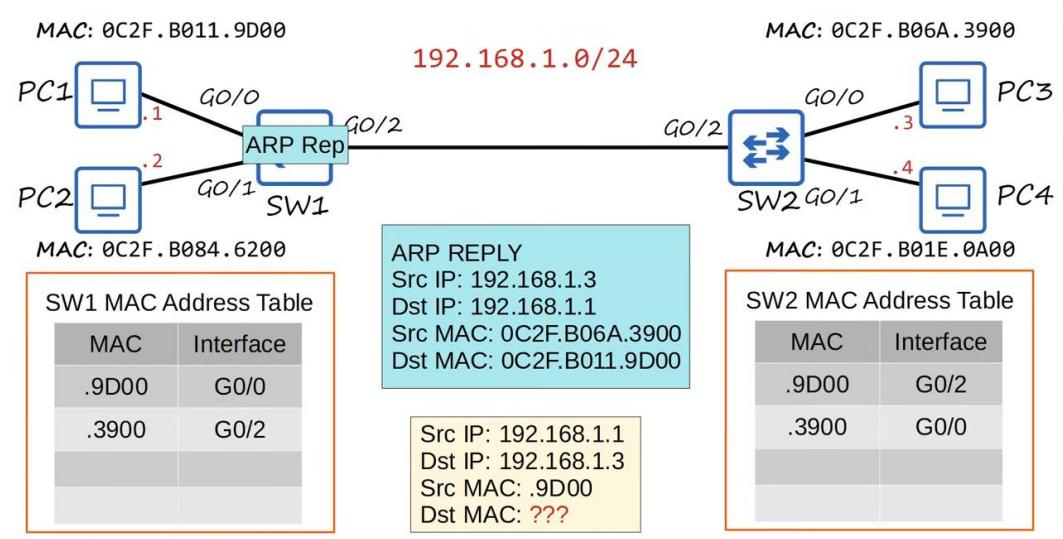
### Ethernet LAN Switching



- When sending data, you only know the IP address of the receiver and not the MAC address
- Switches operate at layer 2 only, so need to know the MAC address
- Address Resolution Protocol
  - Used to discover the MAC address using a known IP address
  - Consists of 2 messages
    - ARP request
    - ARP reply
  - ARP request is broadcast



- ARP reply is unicast



- To view ARP table on pc, use command "arp -a"
  - Internet address - IP address
  - Physical address - MAC address
  - Type static - default entry
  - Type dynamic - learned via ARP

```
C:\Users\user>arp -a

Interface: 169.254.146.29 --- 0x9
  Internet Address        Physical Address      Type
  169.254.255.255        ff-ff-ff-ff-ff-ff    static
  224.0.0.2                01-00-5e-00-00-02    static
  224.0.0.22               01-00-5e-00-00-16    static
  224.0.0.251              01-00-5e-00-00-fb    static
  224.0.0.252              01-00-5e-00-00-fc    static
  239.255.255.250          01-00-5e-7f-ff-fa    static
  255.255.255.255          ff-ff-ff-ff-ff-ff    static

Interface: 192.168.0.167 --- 0xd
  Internet Address        Physical Address      Type
  192.168.0.1              98-da-c4-dd-a8-e4    dynamic
  192.168.0.255             ff-ff-ff-ff-ff-ff    static
  224.0.0.2                01-00-5e-00-00-02    static
  224.0.0.22               01-00-5e-00-00-16    static
  224.0.0.251              01-00-5e-00-00-fb    static
  224.0.0.252              01-00-5e-00-00-fc    static
  239.255.255.250          01-00-5e-7f-ff-fa    static
  255.255.255.255          ff-ff-ff-ff-ff-ff    static
```

- Ping
  - A network utility that is used to test reachability
  - Measures round-trip time
  - Uses 2 messages
    - ICMP Echo Request
    - ICMP Echo Reply
    - Both are unicast, so need to know the MAC address first
  - Command is "ping {ip\_address}"

```
PC1#
PC1#ping 192.168.1.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.3, timeout is 2 seconds:
!!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 20/20/22 ms
PC1#
```

```
PC1#show arp
Protocol Address           Age (min)  Hardware Addr  Type  Interface
Internet 192.168.1.1        -          0c2f.b011.9d00  ARPA  GigabitEthernet0/0
Internet 192.168.1.3       34          0c2f.b06a.3900  ARPA  GigabitEthernet0/0
PC1#
```

- By default, will send 5, 100byte ICMP Echos
- '!': failed ping
- '!': successful ping
- First ping fail because of ARP
  - PC1 does not know the destination MAC address, so need to use ARP, causing the first ping to fail
  - After knowing MAC address, the pings were successful

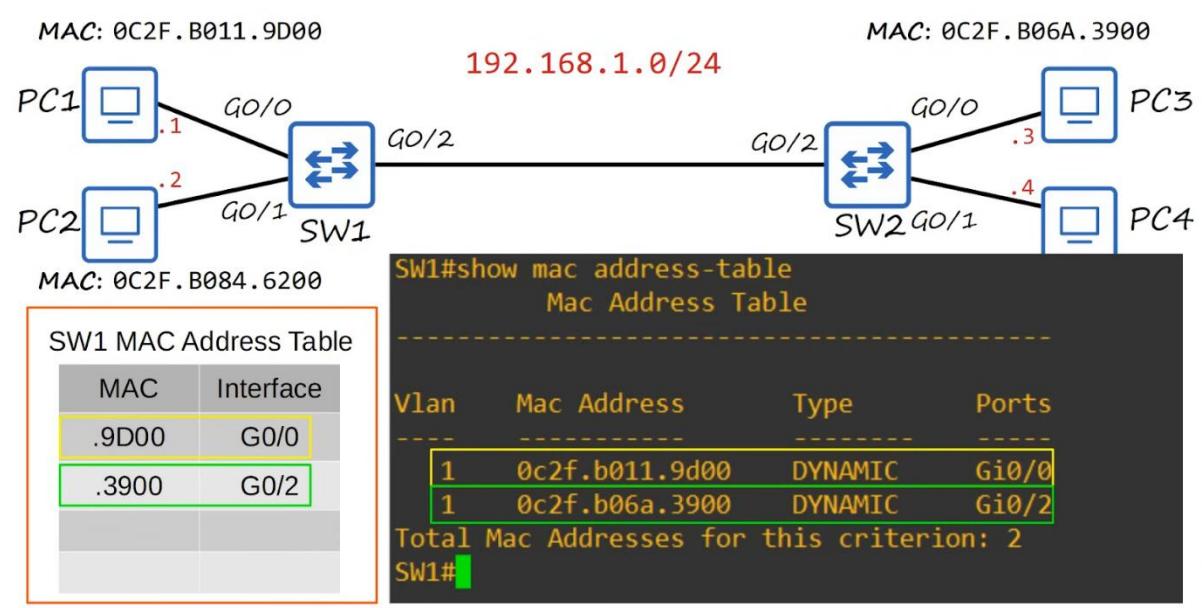
Capturing from - [PC1 Gi0/0 to SW1 Gi0/0]

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	0c:2f:b0:11:9d:00	DEC-MOP-Remote-Cons...	0x6002	77	DEC DNA Remote Console
2	10.593169	0c:2f:b0:11:9d:00	Broadcast	ARP	60	Who has 192.168.1.3? Tell 192.168.1.1
3	10.626235	0c:2f:b0:6a:39:00	0c:2f:b0:11:9d:00	ARP	60	192.168.1.3 is at 0c:2f:b0:6a:39:00
4	12.594539	192.168.1.1	192.168.1.3	ICMP	114	Echo (ping) request id=0x0000, seq=1/256,
5	12.611613	192.168.1.3	192.168.1.1	ICMP	114	Echo (ping) reply id=0x0000, seq=1/256,
6	12.615710	192.168.1.1	192.168.1.3	ICMP	114	Echo (ping) request id=0x0000, seq=2/512,
7	12.635834	192.168.1.3	192.168.1.1	ICMP	114	Echo (ping) reply id=0x0000, seq=2/512,
8	12.638777	192.168.1.1	192.168.1.3	ICMP	114	Echo (ping) request id=0x0000, seq=3/768,
9	12.657810	192.168.1.3	192.168.1.1	ICMP	114	Echo (ping) reply id=0x0000, seq=3/768,
10	12.662283	192.168.1.1	192.168.1.3	ICMP	114	Echo (ping) request id=0x0000, seq=4/1024,
11	12.679631	192.168.1.3	192.168.1.1	ICMP	114	Echo (ping) reply id=0x0000, seq=4/1024,
12	61.223287	0c:2f:b0:84:62:00	DEC-MOP-Remote-Cons...	0x6002	77	DEC DNA Remote Console
13	556.051745	0c:2f:b0:1e:0a:00	DEC-MOP-Remote-Cons...	0x6002	77	DEC DNA Remote Console
44	61.223287	0c:2f:b0:84:62:00	DEC-MOP-Remote-Cons...	0x6002	77	DEC DNA Remote Console

## MAC Address Table

```
SW1#show mac address-table
Mac Address Table
-----
Vlan      Mac Address          Type      Ports
----      -----              -----    -----
  1        0c2f.b011.9d00    DYNAMIC   Gi0/0
  1        0c2f.b06a.3900    DYNAMIC   Gi0/2
Total Mac Addresses for this criterion: 2
SW1#
```

- Command in Cisco CLI "show mac address-table"
- VLAN - default = 1
- Ports = interfaces



### Clearing the MAC Address Table

```

Sw1#show mac address-table
Mac Address Table
-----
Vlan      Mac Address          Type      Ports
----      -----              -----      -----
 1        0c2f.b011.9d00    DYNAMIC   Gi0/0
 1        0c2f.b06a.3900    DYNAMIC   Gi0/2
Total Mac Addresses for this criterion: 2
Sw1#clear mac address-table dynamic
Sw1#show mac address-table
Mac Address Table
-----
Vlan      Mac Address          Type      Ports
----      -----              -----      -----
Sw1#

```

- Ageing: After 5 mins, removed from table

- Use "**clear mac address-table dynamic**" to manually remove all
- OR "**clear mac address-table dynamic address {MAC\_address}**"

```
SW1#show mac address-table
      Mac Address Table
-----
Vlan      Mac Address          Type      Ports
---  -----
 1        0c2f.b011.9d00    DYNAMIC   Gi0/0
 1        0c2f.b06a.3900    DYNAMIC   Gi0/2
Total Mac Addresses for this criterion: 2
SW1#clear mac address-table dynamic address 0c2f.b011.9d00
SW1#show mac address-table
      Mac Address Table
-----
Vlan      Mac Address          Type      Ports
---  -----
 1        0c2f.b06a.3900    DYNAMIC   Gi0/2
Total Mac Addresses for this criterion: 1
SW1#
```

- OR "clear mac address-table dynamic interface {interface-id}"

```

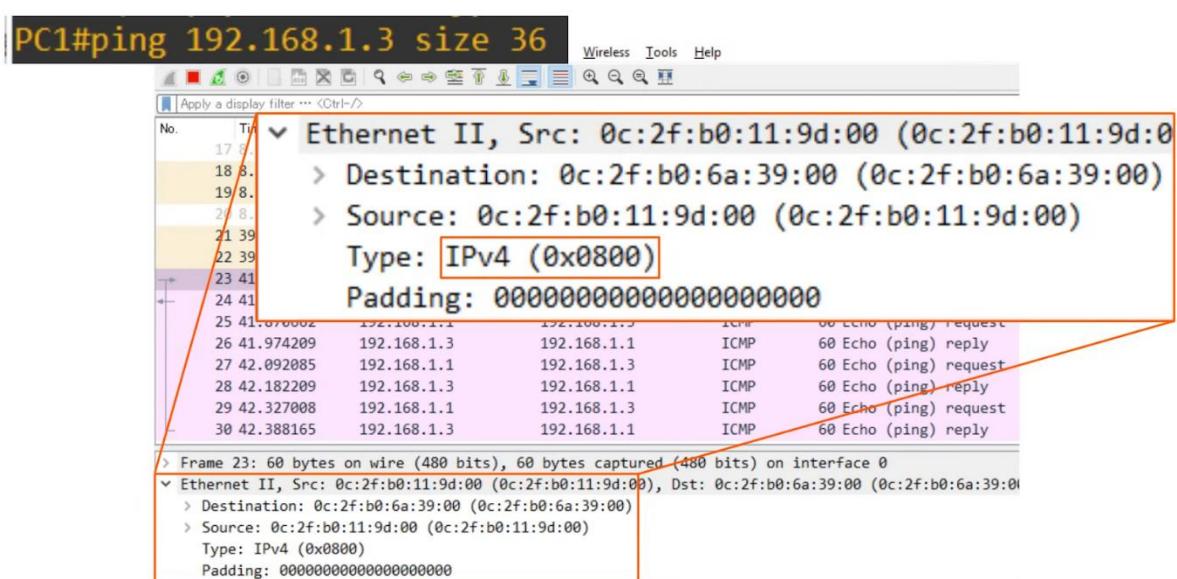
SW1#show mac address-table
      Mac Address Table

-----
Vlan      Mac Address          Type      Ports
----      -----
  1      0c2f.b011.9d00    DYNAMIC    Gi0/0
  1      0c2f.b06a.3900    DYNAMIC    Gi0/2
Total Mac Addresses for this criterion: 2
SW1#clear mac address-table dynamic interface Gi0/0
SW1#show mac address-table
      Mac Address Table

-----
Vlan      Mac Address          Type      Ports
----      -----
  1      0c2f.b06a.3900    DYNAMIC    Gi0/2
Total Mac Addresses for this criterion: 1
SW1#

```

## Ethernet Frame

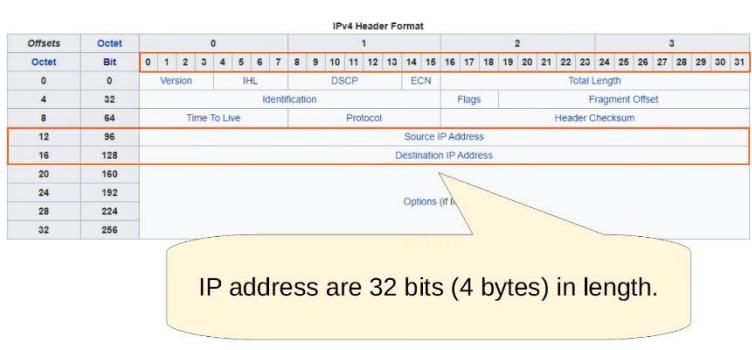


> Frame 22: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface
▼ Ethernet II, Src: 0c:2f:b0:6a:39:00 (0c:2f:b0:6a:39:00), Dst: 0c:2f:b0:11:9d:00 (0c:2f:b0:11:9d:00)
> Destination: 0c:2f:b0:11:9d:00 (0c:2f:b0:11:9d:00)
> Source: 0c:2f:b0:6a:39:00 (0c:2f:b0:6a:39:00)
Type: ARP (0x0806)
Padding: 00
> Address Resolution Protocol (reply)
28 42.182209 192.168.1.3 192.168.1.1 ICMP 60 Echo
29 42.327008 192.168.1.1 192.168.1.3 ICMP 60 Echo
30 42.388165 192.168.1.3 192.168.1.1 ICMP 60 Echo
31 524.651742 0c:2f:b0:1e:0a:00 DEC-MOP-Remote-Cons... 0x6002 77 DEC
32 528.483094 0c:2f:b0:84:62:00 DEC-MOP-Remote-Cons... 0x6002 77 DEC
33 533.098827 0c:2f:b0:11:9d:00 DEC-MOP-Remote-Cons... 0x6002 77 DEC
34 651.573757 0c:2f:b0:6a:39:00 DEC-MOP-Remote-Cons... 0x6002 77 DEC
> Frame 22: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface
▼ Ethernet II, Src: 0c:2f:b0:6a:39:00 (0c:2f:b0:6a:39:00), Dst: 0c:2f:b0:11:9d:00 (0c:2f:b0:11:9d:00)
> Destination: 0c:2f:b0:11:9d:00 (0c:2f:b0:11:9d:00)
> Source: 0c:2f:b0:6a:39:00 (0c:2f:b0:6a:39:00)
Type: ARP (0x0806)
Padding: 00
> Address Resolution Protocol (reply)

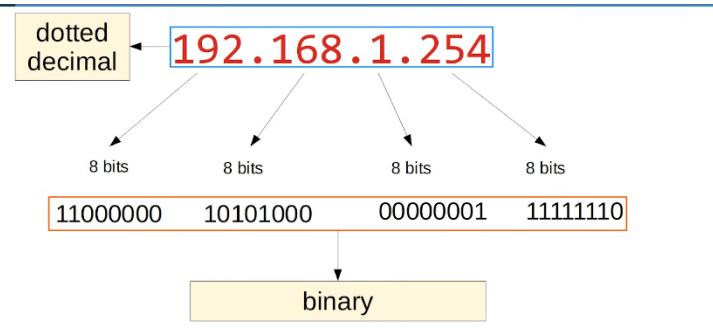
## IPv4 Addressing

### IPv4 Addressing (Part 1)

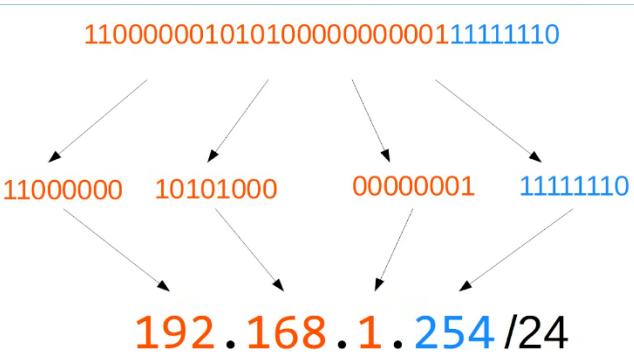
#### IPv4 Header



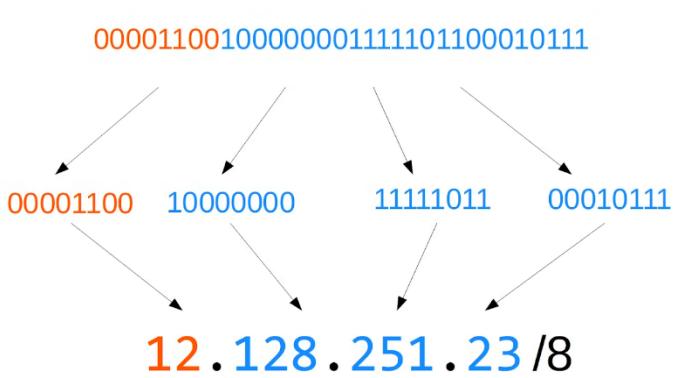
#### IPv4 Address

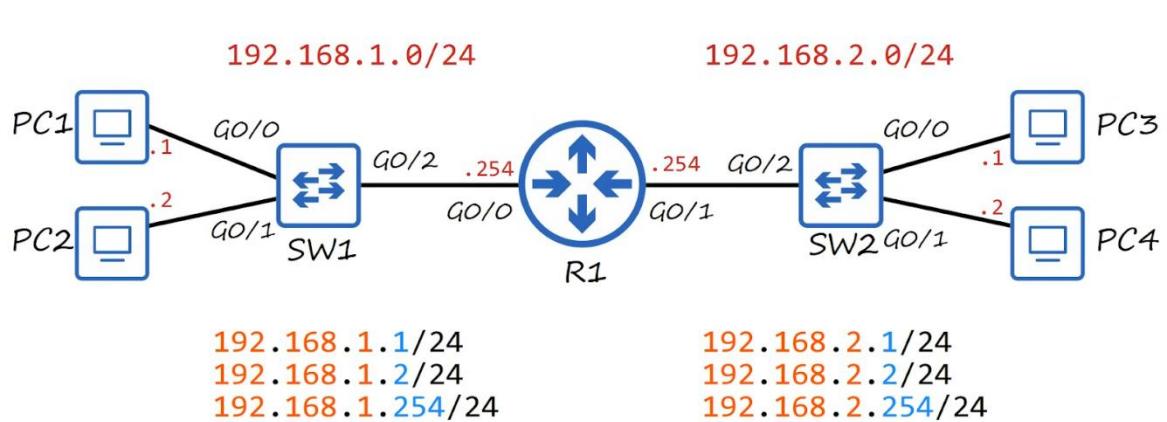


- 'octet': the group of 8 bits
  - So each 8 bits is an octet
- Dotted decimal so that easier for human to read



- The /24 at the end
  - Shows that the first 24 bits of the address refers to the network portion of the address
  - The remaining 8 bits refers to the host portion





Class	First octet	First octet numeric range
A	0xxxxxxx	0-127 <sup>126</sup>
B	10xxxxxx	128-191
C	110xxxxx	192-223
Multicast addresses	D	1110xxxx
Reserved (experimental)	E	1111xxxx

## Loopback Addresses

- Address range 127.0.0.0 - 127.255.255.255
- Used to test the 'network stack' (think OSI, TCP/IP model) on the local device

```
C:\Users\user>ping 127.0.0.1
Pinging 127.0.0.1 with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\user>ping 127.23.68.241
Pinging 127.23.68.241 with 32 bytes of data:
Reply from 127.23.68.241: bytes=32 time<1ms TTL=128

Ping statistics for 127.23.68.241:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

- Pinging your own computer

## IPv4 Address Classes

Class	First octet	First octet numeric range	Prefix Length
A	0xxxxxxx	0-127	/8
B	10xxxxxx	128-191	/16
C	110xxxxx	192-223	/24

Class A: **12.128.251.23/8**

Class B: **154.78.111.32/16**

Class C: **192.168.1.254/24**

- Class A: more host, but less network
- Class C: more networks, but less hosts

Class	Leading bits	Size of <i>network number</i> bit field	Size of <i>rest</i> bit field	Number of networks	Addresses per network
Class A	0	8	24	128 ( $2^7$ )	16,777,216 ( $2^{24}$ )
Class B	10	16	16	16,384 ( $2^{14}$ )	65,536 ( $2^{16}$ )
Class C	110	24	8	2,097,152 ( $2^{21}$ )	256 ( $2^8$ )

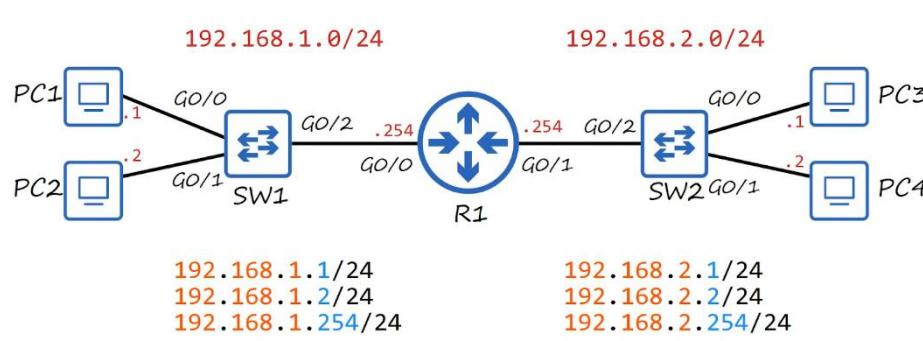
254

- For the last column, the first and last address is reserved
  - First address is the network address
  - Last address is the broadcast address, used to broadcast to all hosts in the network

## Netmask

Class A: /8	$255.0.0.0$ (11111111 00000000 00000000 00000000)
Class B: /16	$255.255.0.0$ (11111111 11111111 00000000 00000000)
Class C: /24	$255.255.255.0$ (11111111 11111111 11111111 00000000)

- They are the same thing represented differently



- Host portion of the address is all 0's - Network address
  - See picture above, 192.168.1.0 and 192.168.2.0 are network addresses
  - Cannot be assigned to a host
- Host portion of the address is all 1's - Broadcast address
  - Cannot be assigned to a host

## Review

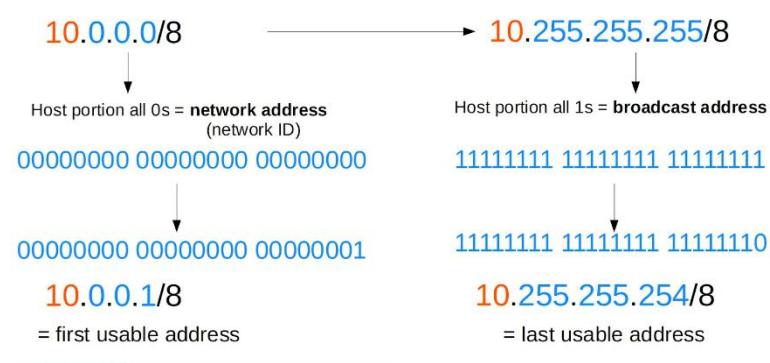
- Dotted decimal and binary
- Network portion/ Host portion of IPv4 address
- IPv4 classes

- Prefix length / netmasks
- Network addresses / Broadcast addresses

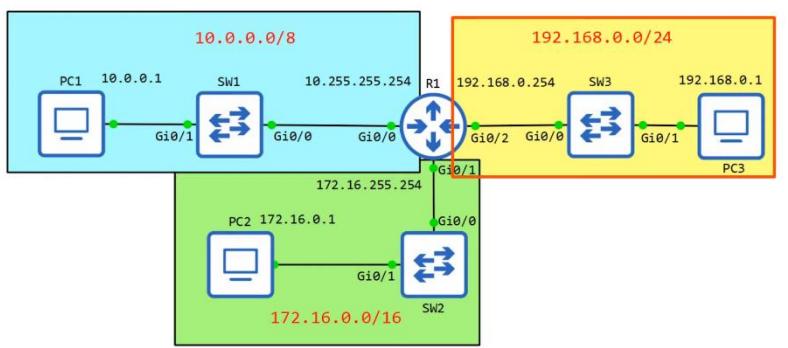
## IPv4 Addressing (Part 2)

### Max Hosts per Network

- Max hosts per network =  $2^n - 2$
- Subtract 2 because
  - Host portion all 0 is network address
  - Host portion all 1 is broadcast address
- First usable address
  - Host portion = 1
  - Host portion = all 1 except LSB



CLI



```
R1>en
R1#show ip interface brief
Interface          IP-Address      OK? Method Status       Protocol
GigabitEthernet0/0 unassigned      YES unset administratively down
down
GigabitEthernet0/1 unassigned      YES unset administratively down
down
GigabitEthernet0/2 unassigned      YES unset administratively down
down
GigabitEthernet0/3 unassigned      YES unset administratively down
down
R1#
```

- "administratively down": interface has been disabled with the "shutdown" command
- Default status for Cisco router interfaces
  - Even when the interface is already connected
- For switches, default is NOT "administratively down"
- "Status" : Layer 1 status (e.g. cable is down)
- "Protocol" : Layer 2 status (e.g. is Ethernet working properly between this device and the device it is connected to)

## Interface Configuration Mode

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#interface gigabitethernet 0/0
R1(config-if)#[
```

R1(config)#interface gigabitethernet0/0 R1(config-if)#[	R1(config)#in g? GMPLS GigabitEthernet Group-Async
R1(config)#i? id-manager ida-client identity interface ip ipc iphc-profile ipv6 isis ixi	R1(config)#in g0/0 R1(config-if)#[
R1(config)#in? interface	

## Configuring IP Address

R1(config-if)#ip address 10.255.255.254 ? A.B.C.D IP subnet mask	
R1(config-if)#ip address 10.255.255.254 255.0.0.0 R1(config-if)#no shutdown R1(config-if)#[br/> *Dec 7 08:29:08.937: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up *Dec 7 08:29:09.938: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up R1(config-if)#[	
R1(config-if)#do sh ip int br Interface IP-Address OK? Method Status Protocol GigabitEthernet0/0 10.255.255.254 YES manual up up GigabitEthernet0/1 unassigned YES unset administratively down down GigabitEthernet0/2 unassigned YES unset administratively down down GigabitEthernet0/3 unassigned YES unset administratively down down R1(config-if)#[	

R1(config-if)#int g0/1 R1(config-if)#ip add 172.16.255.254 255.255.0.0 R1(config-if)#no shut R1(config-if)#[br/> *Dec 7 08:51:42.648: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to up *Dec 7 08:51:43.649: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to up R1(config-if)#do sh ip int br Interface IP-Address OK? Method Status Protocol GigabitEthernet0/0 10.255.255.254 YES manual up up GigabitEthernet0/1 172.16.255.254 YES manual up up GigabitEthernet0/2 unassigned YES unset administratively down down GigabitEthernet0/3 unassigned YES unset administratively down down R1(config-if)#[	
--	--

```

R1(config-if)#int g0/2
R1(config-if)#ip add 192.168.0.254 255.255.255.0
R1(config-if)#no shut
R1(config-if)#
*Dec  7 09:05:41.505: %LINK-3-UPDOWN: Interface GigabitEthernet0/2, changed state to up
R1(config-if)#
*Dec  7 09:05:42.505: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/2, changed state to up
R1(config-if)#do sh ip int br
Interface          IP-Address      OK? Method Status      Protocol
GigabitEthernet0/0  10.255.255.254 YES manual up        up
GigabitEthernet0/1  172.16.255.254 YES manual up        up
GigabitEthernet0/2  192.168.0.254 YES manual up        up
GigabitEthernet0/3  unassigned      YES unset administratively down down
R1(config-if)#

```

## More "show" commands

```

R1#show interfaces g0/0
GigabitEthernet0/0 is up, line protocol is up
  Hardware is iGbE, address is 0c1b.8444.f000 (bia 0c1b.8444.f000)
  Internet address is 10.255.255.254/8
  MTU 1500 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Auto Duplex, Auto Speed, link type is auto, media type is RJ45
  output flow-control is unsupported, input flow-control is unsupported
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:06, output 00:00:05, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    167 packets input, 30159 bytes, 0 no buffer
    Received 0 broadcasts (0 IP multicasts)
    0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
    0 watchdog, 0 multicast, 0 pause input
    350 packets output, 39097 bytes, 0 underruns
    0 output errors, 0 collisions, 2 interface resets
    105 unknown protocol drops
    0 babbles, 0 late collision, 0 deferred
    1 lost carrier, 0 no carrier, 0 pause output
    0 output buffer failures, 0 output buffers swapped out

```

```
R1#show interfaces description
Interface                  Status      Protocol Description
Gi0/0                     up          up
Gi0/1                     up          up
Gi0/2                     up          up
Gi0/3                     admin down  down
```

```
R1(config)#int g0/0
R1(config-if)#description ## to SW1 ##
R1(config-if)#int g0/1
R1(config-if)#desc ## to SW2 ##
R1(config-if)#int g0/2
R1(config-if)#desc ## to SW3 ##
R1(config-if)#do sh int desc
Interface                  Status      Protocol Description
Gi0/0                     up          up      ## to SW1 ##
Gi0/1                     up          up      ## to SW2 ##
Gi0/2                     up          up      ## to SW3 ##
Gi0/3                     admin down  down
```

## Topics Covered

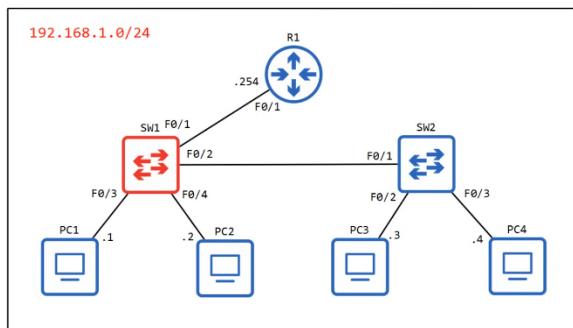
- IPv4 address classes (review, clarification)
- Finding , for a particular network
  - Max number of hosts
  - Network address
  - Broadcast address
  - First useable address
  - Last useable address
- Configuring IP addresses on Cisco devices

## Commands

- **show ip interface brief** (privileged mode)

- To set IP address
  - **interface {interface}** (e.g. gigabitethernet0/0 OR g0/0, global mode)
  - **ip address {ip address}{subnet mask}** (interface mode)
  - **no shutdown** (interface mode)
  - **description {description}** (interface mode)
- **show interfaces {interface}** (privileged mode)
- **show interfaces description** (privileged mode)

## Switch Interfaces



Interface	IP-Address	OK?	Method	Status	Protocol
Vlan 1	unassigned	YES	unset	up	up
FastEthernet0/1	unassigned	YES	unset	up	up
FastEthernet0/2	unassigned	YES	unset	up	up
FastEthernet0/3	unassigned	YES	unset	up	up
FastEthernet0/4	unassigned	YES	unset	up	up
FastEthernet0/5	unassigned	YES	unset	down	down
FastEthernet0/6	unassigned	YES	unset	down	down
FastEthernet0/7	unassigned	YES	unset	down	down
FastEthernet0/8	unassigned	YES	unset	down	down
FastEthernet0/9	unassigned	YES	unset	down	down
FastEthernet0/10	unassigned	YES	unset	down	down
FastEthernet0/11	unassigned	YES	unset	down	down
FastEthernet0/12	unassigned	YES	unset	down	down

- By default, switches NOT shutdown unlike routers
  - Will either be up/up if connected
  - Down/down if not connected
- Difference between "down" and "administratively down"
  - "down": not connected to anything

- o "admin down": 'shutdown' command applied

SW1#show interfaces status						
Port	Name	Status	Vlan	Duplex	Speed	Type
Fa0/1		connected	1	a-full	a-100	10/100BaseTX
Fa0/2		connected	trunk	a-full	a-100	10/100BaseTX
Fa0/3		connected	1	a-full	a-100	10/100BaseTX
Fa0/4		connected	1	a-full	a-100	10/100BaseTX
Fa0/5		notconnect	1	auto	auto	10/100BaseTX
Fa0/6		notconnect	1	auto	auto	10/100BaseTX
Fa0/7		notconnect	1	auto	auto	10/100BaseTX
Fa0/8		notconnect	1	auto	auto	10/100BaseTX
Fa0/9		notconnect	1	auto	auto	10/100BaseTX
Fa0/10		notconnect	1	auto	auto	10/100BaseTX
Fa0/11		notconnect	1	auto	auto	10/100BaseTX
Fa0/12		notconnect	1	auto	auto	10/100BaseTX

- Duplex
  - o "a-full": auto full duplex

Configure Speed & Duplex

```
Sw1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Sw1(config)#int f0/1
Sw1(config-if)#speed ?
10                      Force 10 Mbps operation
100                     Force 100 Mbps operation
auto                    Enable AUTO speed configuration
Sw1(config-if)#speed 100
Sw1(config-if)#duplex ?
auto                    Enable AUTO duplex configuration
full                   Force full duplex operation
half                  Force half-duplex operation
Sw1(config-if)#duplex full
Sw1(config-if)#description ## to R1 ##
```

Port	Name	Status	Vlan	Duplex	Speed	Type
Fa0/1	## to R1 ##	connected	1	full	100	10/100BaseTX
Fa0/2		connected	trunk	a-full	a-100	10/100BaseTX
Fa0/3		connected	1	a-full	a-100	10/100BaseTX
Fa0/4		connected	1	a-full	a-100	10/100BaseTX
Fa0/5		notconnect	1	auto	auto	10/100BaseTX
Fa0/6		notconnect	1	auto	auto	10/100BaseTX
Fa0/7		notconnect	1	auto	auto	10/100BaseTX
Fa0/8		notconnect	1	auto	auto	10/100BaseTX
Fa0/9		notconnect	1	auto	auto	10/100BaseTX
Fa0/10		notconnect	1	auto	auto	10/100BaseTX
Fa0/11		notconnect	1	auto	auto	10/100BaseTX
Fa0/12		notconnect	1	auto	auto	10/100BaseTX

### Shutdown Not-used Interface

```
SW1(config)#interface range f0/5 - 12
SW1(config-if-range)#description ## not in use ##
SW1(config-if-range)#shutdown
00:42:36: %LINK-5-CHANGED: Interface FastEthernet0/5, changed state to administratively down
00:42:36: %LINK-5-CHANGED: Interface FastEthernet0/6, changed state to administratively down
00:42:36: %LINK-5-CHANGED: Interface FastEthernet0/7, changed state to administratively down
00:42:36: %LINK-5-CHANGED: Interface FastEthernet0/8, changed state to administratively down
00:42:36: %LINK-5-CHANGED: Interface FastEthernet0/9, changed state to administratively down
00:42:36: %LINK-5-CHANGED: Interface FastEthernet0/10, changed state to administratively down
00:42:36: %LINK-5-CHANGED: Interface FastEthernet0/11, changed state to administratively down
00:42:36: %LINK-5-CHANGED: Interface FastEthernet0/12, changed state to administratively down
SW1(config-if-range)#

```

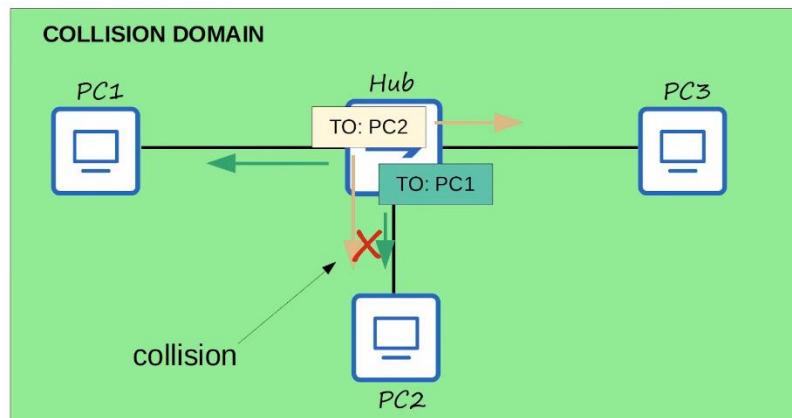
```
SW1(config)#int range f0/5 - 6, f0/9 - 12
SW1(config-if-range)#no shut
00:57:07: %LINK-3-UPDOWN: Interface FastEthernet0/5, changed state to up
00:57:07: %LINK-3-UPDOWN: Interface FastEthernet0/6, changed state to up
00:57:07: %LINK-3-UPDOWN: Interface FastEthernet0/9, changed state to up
00:57:07: %LINK-3-UPDOWN: Interface FastEthernet0/10, changed state to up
00:57:07: %LINK-3-UPDOWN: Interface FastEthernet0/11, changed state to up
00:57:07: %LINK-3-UPDOWN: Interface FastEthernet0/12, changed state to up

```

Port	Name	Status	Vlan	Duplex	Speed	Type
Fa0/1	## to R1 ##	connected	1	full	100	10/100BaseTX
Fa0/2	## to SW2 ##	connected	trunk	a-full	a-100	10/100BaseTX
Fa0/3	## to end hosts ##	connected	1	a-full	a-100	10/100BaseTX
Fa0/4	## to end hosts ##	connected	1	a-full	a-100	10/100BaseTX
Fa0/5	## not in use ##	disabled	1	auto	auto	10/100BaseTX
Fa0/6	## not in use ##	disabled	1	auto	auto	10/100BaseTX
Fa0/7	## not in use ##	disabled	1	auto	auto	10/100BaseTX
Fa0/8	## not in use ##	disabled	1	auto	auto	10/100BaseTX
Fa0/9	## not in use ##	disabled	1	auto	auto	10/100BaseTX
Fa0/10	## not in use ##	disabled	1	auto	auto	10/100BaseTX
Fa0/11	## not in use ##	disabled	1	auto	auto	10/100BaseTX
Fa0/12	## not in use ##	disabled	1	auto	auto	10/100BaseTX

- Full duplex
  - Device can send and receive at the same time
- Half duplex
  - Can only send or receive at the same time

## Hub

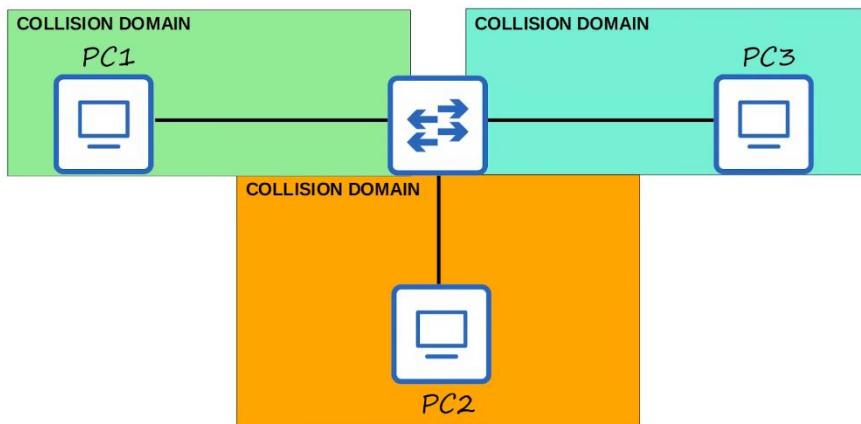


- Hub is like a repeater, sends out message received to everyone
- Only the destination will recognise and accept, the rest will ignore
- Collision can occur when hub transmit multiple messages at the same time

## CSMA/CD

- Carrier Sense Multiple Access / Collision Detection
- Before sending, devices listen to collision domain until they detect that other devices are not sending
- If a collision occur, the device sends a jamming signal to inform the other devices that a collision occurred
- Each device will wait a random period of time before sending frames again
- The process repeats

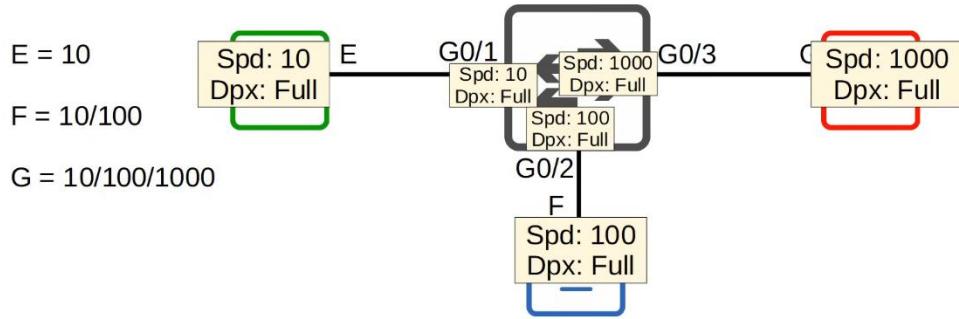
## Switch Collision Domain



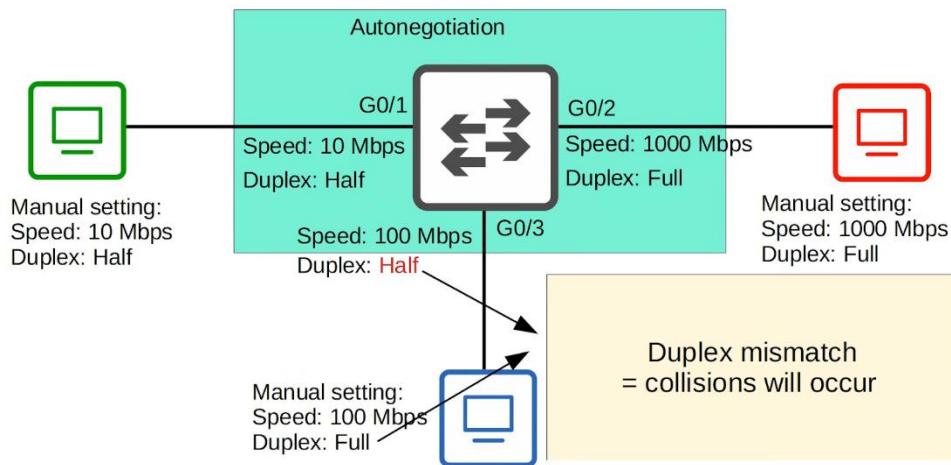
- Less likely for collision to occur
- If occur, likely due to configuration

## Speed/Duplex Auto-negotiation

- Interfaces that can run at different speeds (10/100 or 10/100/1000) have default settings of 'speed auto' and 'duplex auto'
- Interfaces 'advertise' their capabilities to the neighbouring device, and they negotiate the best speed and duplex settings they are both capable of



- What if autonegotiation is disabled?
  - Speed
    - The switch will try to sense the speed that the other device is operating at
    - If cannot, it will use the slowest supported speed
  - Duplex
    - If speed is 10 or 100 Mbps, half duplex
    - If 1000 or more, full duplex



```

SW1#show interfaces f0/2
FastEthernet0/2 is up, line protocol is up
  Hardware is Fast Ethernet, address is 000C.3168.8461 (bia 000C.3168.8461)
  Description: ## to SW2 ##
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Auto-duplex, Auto-speed
  Encapsulation ARPA, loopback not set
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 02:29:44, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queuing strategy: fifo
  Output queue :0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    269 packets input, 71059 bytes, 0 no buffer
    Received 6 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
    7290 packets output, 429075 bytes, 0 underruns
    0 output errors, 3 interface resets
    0 output buffer failures, 0 output buffers swapped out

```

- Runts
  - Frames that are smaller than the min Ethernet frame size (64 bytes)
- Giants
  - Frames that are larger than the max Ethernet frame size (1518 bytes)
- CRC
  - Frames that failed the CRC check (in the Ethernet FCS trailer)
- Frame
  - Frames that have an incorrect format (due to an error)
- Input errors
  - Total of various counters, such as the above four
- Output errors
  - Frames the switch tried to send, but failed due to an error

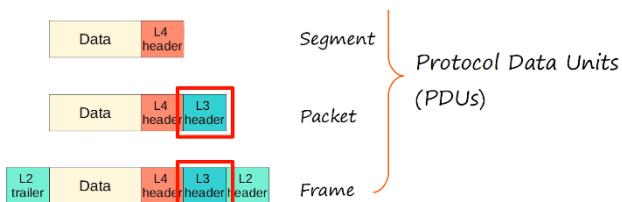
## Review

- Interface speed & duplex
- Speed & duplex auto negotiation
- Interface status
- Interface counters & errors

## IPv4 Header

### Content

- IPv4 Structure
- Fields of the IPv4 header



### Header

Offsets	Octet	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31																	
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31																	
0	0	Version		IHL		DSCH		ECN		Total Length																																								
4	32	Identification										Flags		Fragment Offset																																				
8	64	Time To Live				Protocol				Header Checksum																																								
12	96	Source IP Address																																																
16	128	Destination IP Address																																																
20	160																																																	
24	192																																																	
28	224	Options (if IHL > 5)																																																
32	256																																																	

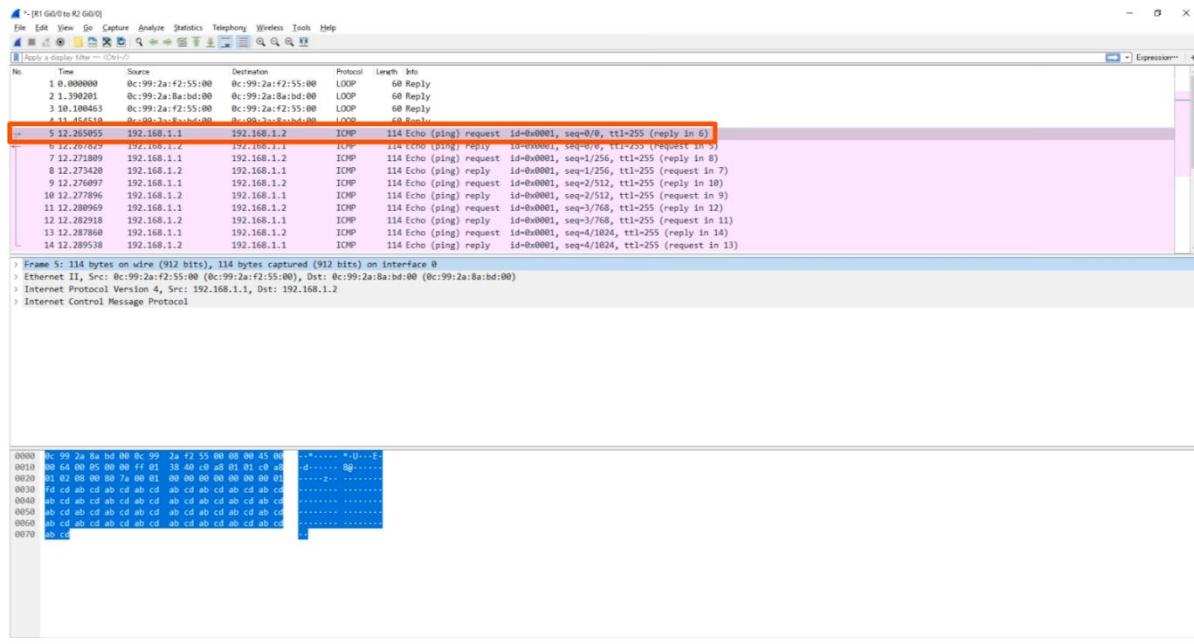
- Version

- 4 bits
  - Indicates version of IP used
  - IPv4 = 0100 (4)
  - IPv6 = 0110 (6)
- **Internet Header Length (IHL)**
  - 4 bits
  - The final field of the IPv4 header (Options) is variable in length, so this field is necessary to indicate the total length of the header
  - Identifies the length of the header **in 4-bytes increment**
  - Value of 5 =  $5 * 4$  bytes = 20bytes
  - Min value is 5 = 20 bytes
  - Max value is (1111) 15 = 60 bytes
- **Differentiated Services Code Point (DSCP)**
  - 6 bits
  - Used for QoS (quality of service)
  - Used to prioritize delay-sensitive data (streaming voice/video etc)
- **Explicit Congestion Notification (ECN)**
  - 2 bits
  - Provides end-to-end (btw 2 endpoints) notification of network congestion **without dropping packets**
  - Optional feature that requires both endpoints and the underlying infrastructure to support it
- **Total Length**
  - 16 bits
  - Indicates the total length of the packet (L3 + L4 segment)
  - Measured in bytes (NOT 4 bytes increment)
  - Min value 20 (IPv4 header with no encapsulated data)
  - Max value 65,535 (16 bits all 1)
- **Identification Field**

- 16 bits
  - If a packet too large and is fragmented, this field used to identify which packet the fragment belongs to
  - All fragments of the same packet will have their own IPv4 header with the same value in this field
  - Packets are fragmented if larger than the MTU (Max Transmission Unit)
  - MTU usually 1500 bytes (max size of Ethernet frame)
  - Fragments are reassembled by the receiving host
- **Flags field**
    - 3 bits
    - Used to control/identify fragments
    - Bit 0: Reserved, always set to 0
    - Bit 1: Don't fragment (DF bit), used to indicate a packet that should not be fragmented
    - Bit 2: More fragments (MF bit), set to 1 if there are more fragments in the packet, set to 0 for the last fragment
    - Unfragmented packets will always have their MF bit set to 0
  - **Fragment Offset**
    - 13 bits
    - Used to indicate the position of the fragment within the original, unfragmented IP packet
    - Allow fragmented packets to be reassembled even if the fragments arrive out of order
  - **Time To Live**
    - 8 bits
    - A router will drop a packet with a TTL of 0
    - Used to prevent infinite loops
    - Originally designed to indicate the packet's max lifetime in seconds
    - In practice, indicates a 'hop count' - each time the packet arrives at a router, the router decreases the TTL by 1

- Recommended default TTL = 64
- **Protocol**
  - 8 bits
  - Indicates the protocol of the encapsulated L4 PDU
  - Value of 6: TCP
  - Value of 17: UDP
  - Value of 1: ICMP
  - Value of 89: OSPF (dynamic routing protocol)
- **Header Checksum**
  - 16 bits
  - A calculated checksum used to check for errors in the IPv4 header
  - When a router receives a packet, it calculates the checksum of the header and compares it to the one in this field of the header
  - If they do not match, the router drops the packet
  - Used to check for error in IPv4 header only
  - IP relies on the encapsulated protocol to detect errors in the encapsulated data
  - Both TCP and UDP have their own checksum fields to detect errors in the encapsulated data
- **Source/Destination IP Address**
  - 32 bits
  - IPv4 address of sender and receiver
- **Options**
  - 0 - 320 bits
  - Rarely used
  - If the IHL field is greater than 5, it means that Options are present

WireShark



▼ Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1.2

0100 .... = Version: 4

.... 0101 = Header Length: 20 bytes (5)

▼ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

0000 00.. = Differentiated Services Codepoint: Default (0)

.... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)

Total Length: 100

Identification: 0x0005 (5)

▼ Flags: 0x0000

0... .... .... .... = Reserved bit: Not set

.0... .... .... .... = Don't fragment: Not set

..0. .... .... .... = More fragments: Not set

...0 0000 0000 0000 = Fragment offset: 0

Time to live: 255

Protocol: ICMP (1)

Header checksum: 0x3840 [validation disabled]

[Header checksum status: Unverified]

Source: 192.168.1.1

Destination: 192.168.1.2

## R1#ping 192.168.1.2 size 10000

7 17.411175	192.168.1.1	192.168.1.2	IPv4	1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=0001) [Reassembled in #13]
8 17.412827	192.168.1.1	192.168.1.2	IPv4	1514 Fragmented IP protocol (proto=ICMP 1, off=1480, ID=0001) [Reassembled in #13]
9 17.414347	192.168.1.1	192.168.1.2	IPv4	1514 Fragmented IP protocol (proto=ICMP 1, off=2960, ID=0001) [Reassembled in #13]
10 17.415913	192.168.1.1	192.168.1.2	IPv4	1514 Fragmented IP protocol (proto=ICMP 1, off=4440, ID=0001) [Reassembled in #13]
11 17.417560	192.168.1.1	192.168.1.2	IPv4	1514 Fragmented IP protocol (proto=ICMP 1, off=5920, ID=0001) [Reassembled in #13]
12 17.419203	192.168.1.1	192.168.1.2	IPv4	1514 Fragmented IP protocol (proto=ICMP 1, off=7400, ID=0001) [Reassembled in #13]
13 17.420793	192.168.1.1	192.168.1.2	ICMP	1134 Echo (ping) request id=0x0000, seq=1/256, ttl=255 (reply in 20)

Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1.2 0100 .... = Version: 4 .... 0101 = Header Length: 20 bytes (5) ▼ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT) 0000 00.. = Differentiated Services Codepoint: Default (0) ..0 = Explicit Congestion Notification: Not ECN-Capable Transport (0) Total Length: 1500 Identification: 0x0001 (1) ▼ Flags: 0x2000, More fragments 0... .... .... = Reserved bit: Not set .0... .... .... = Don't fragment: Not set ..1. .... .... = More fragments: Set ...0 0000 0000 0000 = Fragment offset: 0 Time to live: 255 Protocol: ICMP (1) Header checksum: 0x12cc [validation disabled] [Header checksum status: Unverified] Source: 192.168.1.1 Destination: 192.168.1.2 Reassembled IPv4 in frame: 13	Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1.2 0100 .... = Version: 4 .... 0101 = Header Length: 20 bytes (5) ▼ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT) 0000 00.. = Differentiated Services Codepoint: Default (0) ..0 = Explicit Congestion Notification: Not ECN-Capable Transport (0) Total Length: 1500 Identification: 0x0001 (1) ▼ Flags: 0x20b9, More fragments 0... .... .... = Reserved bit: Not set .0... .... .... = Don't fragment: Not set ..1. .... .... = More fragments: Set ...0 0000 1011 1001 = Fragment offset: 185 Time to live: 255 Protocol: ICMP (1) Header checksum: 0x1213 [validation disabled] [Header checksum status: Unverified] Source: 192.168.1.1 Destination: 192.168.1.2 Reassembled IPv4 in frame: 13
--	--

## R1#ping 192.168.1.2 df-bit

▼ Flags: 0x4000, Don't fragment  
  0.... .... .... = Reserved bit: Not set  
  .1.. .... .... .... = Don't fragment: Set  
  ..0. .... .... .... = More fragments: Not set  
  ...0 0000 0000 0000 = Fragment offset: 0

```
R1#ping 192.168.1.2 size 10000 df-bit
Type escape sequence to abort.
Sending 5, 10000-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
Packet sent with the DF bit set
.....
Success rate is 0 percent (0/5)
```

# Routing Fundamentals

## Routing Fundamentals

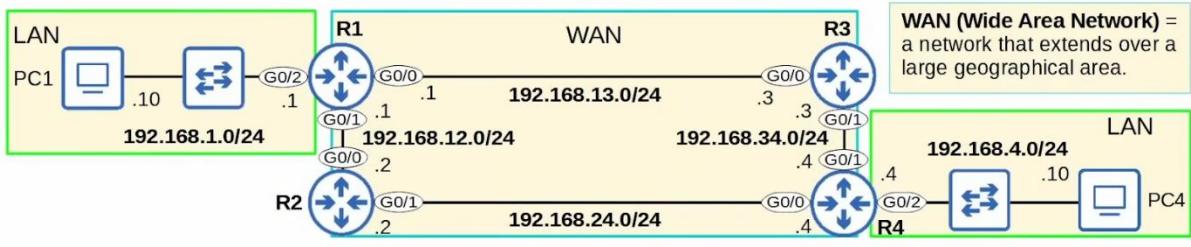
### Topics

- What is routing?
- Routing table on Cisco routers
  - Connected routes
  - Local routes
- Routing fundamentals (route selection)

### What is routing?

- Routing is the process that routers use to determine the path that the IP packets should take over a network to reach their destination
  - Routers store routes to all of their known destinations in a **routing table**
  - When routers receive packets, they look in the **routing table** to find the best route to forward the packets
- 2 main routing methods (ways to learn the route)
  - **Dynamic Routing**
    - Routers use dynamic routing protocols (e.g. OSPF) to share routing information with each other automatically and build their routing tables
  - **Static Routing**
    - Manually configure routes on router
- A **route** tells the router
  - To send a packet to destination X, you should send the packet to **next-hop Y**
    - '**next-hop**

- OR if the destination is directly connected to the router, to send it directly to the destination
- OR if the destination is the router's own IP address, receive the packet for yourself (don't forward it)



```
R1# conf t
R1(config)# interface g0/0
R1(config-if)# ip address 192.168.13.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# interface g0/1
R1(config-if)# ip address 192.168.12.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# interface g0/2
R1(config-if)# ip address 192.168.1.1 255.255.255.0
R1(config-if)# no shutdown

R1# show ip int br
Interface          IP-Address      OK? Method Status           Protocol
GigabitEthernet0/0  192.168.13.1   YES manual up            up
GigabitEthernet0/1  192.168.12.1   YES manual up            up
GigabitEthernet0/2  192.168.1.1    YES manual up            up
GigabitEthernet0/3  unassigned     YES NVRAM administratively down down
```

There is no need to use **exit** to return to global config mode before entering **interface g0/1**. You can use the **interface g0/1** command directly from interface config mode.

'show ip route'

```
R1# show ip route
Use the command show ip route to view the routing table.

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from Pfr

Gateway of last resort is not set

C 192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
   192.168.1.0/24 is directly connected, GigabitEthernet0/2
   192.168.1.1/32 is directly connected, GigabitEthernet0/2
C 192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
   192.168.12.0/24 is directly connected, GigabitEthernet0/1
   192.168.12.1/32 is directly connected, GigabitEthernet0/1
C 192.168.13.0/24 is variably subnetted, 2 subnets, 2 masks
   192.168.13.0/24 is directly connected, GigabitEthernet0/0
   192.168.13.1/32 is directly connected, GigabitEthernet0/0
```

The Codes legend in the output of **show ip route** lists the different protocols which routers can use to learn routes.

- **L - local**  
→ A route to the actual IP address configured on the interface. (with a /32 netmask)
- **C - connected**  
→ A route to the network the interface is connected to. (with the actual netmask configured on the interface)

When you configure an IP address on an interface and enable it with **no shutdown**, 2 routes (per interface) will automatically be added to the routing table:

- a **connected** route
- a **local** route

- Local and Connected routes are not part of dynamic/static routing
  - They are automatically added

## Connected and Local Routes

```
C 192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C   192.168.1.0/24 is directly connected, GigabitEthernet0/2
L   192.168.1.1/32 is directly connected, GigabitEthernet0/2
C   192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C     192.168.12.0/24 is directly connected, GigabitEthernet0/1
L     192.168.12.1/32 is directly connected, GigabitEthernet0/1
C   192.168.13.0/24 is variably subnetted, 2 subnets, 2 masks
C     192.168.13.0/24 is directly connected, GigabitEthernet0/0
L     192.168.13.1/32 is directly connected, GigabitEthernet0/0
```

- A **connected** route is a route to the network the interface is connected to.
- R1 G0/2 IP = **192.168.1.1/24**
- Network Address = **192.168.1.0/24**
- It provides a route to all hosts in that network (ie. **192.168.1.10**, **192.168.1.100**, **192.168.1.232**, etc.)
- R1 knows: "If I need to send a packet to any host in the 192.168.1.0/24 network, I should send it out of G0/2".
- A **local** route is a route to the exact IP address configured on the interface.
- A /32 netmask is used to specify the exact IP address of the interface.  
→ /32 means all 32 bits are 'fixed', they can't change.
- Even though R1's G0/2 is configured as **192.168.1.1/24**, the connected route is to **192.168.1.1/32**.
- R1 knows: "If I receive a packet destined for this IP address, the message is for me".

**192 . 168 . 1 . 0 /24**  
**255 . 255 . 255 . 0**

=**FIXED** (can't change)

```
C 192.168.1.0/24 is directly connected, GigabitEthernet0/2
```

- **192.168.1.0/24** matches 192.168.1.0 ~ 192.168.1.255.  
→ If R1 receives a packet with a destination in that range, it will send the packet out of G0/2.

A route **matches** a packet's destination if the packet's destination IP address is part of the network specified in the route.

=**not fixed**

**192.168.1.2** = **match**

→ Send packet out of G0/2

**192.168.1.7** = **match**

→ Send packet out of G0/2

**192.168.1.89** = **match**

→ Send packet out of G0/2

**192.168.2.1** = **no match**

→ Send the packet using a different route, or drop the packet if there is no matching route.

**192 . 168 . 1 . 1 /32**  
**255 . 255 . 255 . 255**

=**FIXED** (can't change)

192.168.1.1/32 matches ONLY 192.168.1.1

## Route Selection

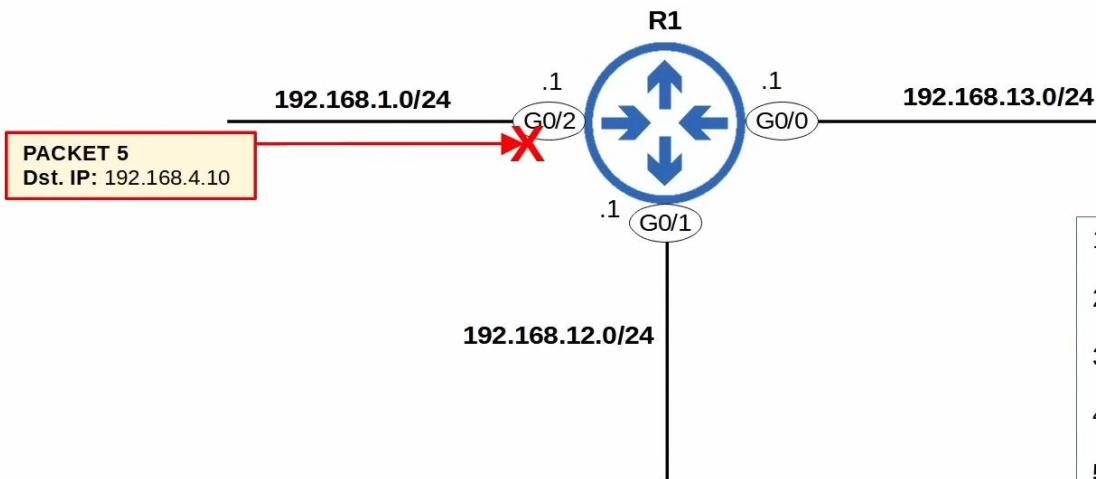
- If R1 receives a packet destined for 192.168.1.1, it will be matched by both Local & Connected routes
  - **192.168.1.0/24** (connected)
  - **192.168.1.1/32** (local)
- The **most specific** route will be chosen
  - Local will be chosen
- **Most specific** matching route = the matching route with the longest prefix
- R1 will receive the packet for itself, rather than sending it out
- Local route = keep the packet, don't forward

```
C 192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
L   192.168.1.0/24 is directly connected, GigabitEthernet0/2
C   192.168.1.1/32 is directly connected, GigabitEthernet0/2
L 192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C   192.168.12.0/24 is directly connected, GigabitEthernet0/1
L   192.168.12.1/32 is directly connected, GigabitEthernet0/1
C 192.168.13.0/24 is variably subnetted, 2 subnets, 2 masks
C   192.168.13.0/24 is directly connected, GigabitEthernet0/0
L   192.168.13.1/32 is directly connected, GigabitEthernet0/0
```

- "**192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks**"
  - In the routing table, there are 2 routes to subnets that fit within the 192.168.1.0/24 class C network, with 2 different netmasks (/24 and /32)
  - Generally can ignore them and just take note of the actual routes

## Route Selection Practice

```
C 192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
L   192.168.1.0/24 is directly connected, GigabitEthernet0/2
L   192.168.1.1/32 is directly connected, GigabitEthernet0/2
C 192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C   192.168.12.0/24 is directly connected, GigabitEthernet0/1
L   192.168.12.1/32 is directly connected, GigabitEthernet0/1
C 192.168.13.0/24 is variably subnetted, 2 subnets, 2 masks
C   192.168.13.0/24 is directly connected, GigabitEthernet0/0
L   192.168.13.1/32 is directly connected, GigabitEthernet0/0
```



- Destination IP: 192.168.1.1
  - Receive for myself
- Destination IP: 192.168.13.3
  - Send to destination (connected to g0/0)
- Destination IP: 192.168.1.244
  - Send to destination (connected to g0/2)
- Destination IP: 192.168.12.1
  - Receive for myself
- Destination IP: 192.168.4.10
  - Drop (no route)

## Summary

- Routers store information about destinations they know in their **routing table**.
  - When they receive packets, they look in the routing table to find the best route to forward the packet.
- Each **route** in the routing table is an instruction:
  - To reach destinations in network X, send the packet to **next-hop** Y (the next router in the path to the destination).
  - If the destination is directly connected (**Connected** route) send the packet directly to the destination.
  - If the destination is your own IP address (**Local** route), receive the packet for yourself.

\*We will look at how **next-hops** work in the next video on **static routes**.
- When you configure an IP address on an interface and enable the interface, two routes are automatically added to the routing table:
  - Connected** route (code **C** in the routing table): A route to the network connected to the interface.
    - ie. if the interface's IP is **192.168.1.1/24**, the route will be to **192.168.1.0/24**.
    - Tells the router: "To send a packet to a destination in this network, send it out of the interface specified in the route".
  - Local** route (code **L** in the routing table): A route to the exact IP address configured on the interface.
    - ie. if the interface's IP is **192.168.1.1/24**, the route will be to **192.168.1.1/32**.
    - Tells the router: "Packets to this destination are for you. You should receive them for yourself (not forward them)".
- A route **matches** a destination if the packet's destination IP address is part of the network specified in the route.
  - ie. a packet to **192.168.1.60** is matched by a route to **192.168.1.0/24**, but not by a route to **192.168.0.0/24**.
- If a router receives a packet and it doesn't have a route that matches the packet's destination, it will **drop** the packet.
  - This is different than switches, which **flood** frames if they don't have a MAC table entry for the destination.
- If a router receives a packet and it has multiple routes that match the packet's destination, it will use the **most specific matching route** to forward the packet.
  - **Most specific** matching route = the matching route with the longest prefix length.
  - This is different than switches, which look for an **exact** match in the MAC address table to forward frames.

## Static Routing

### Topics

- Review: Connected and Local routes
- Intro to static routes
- Static Route config
- Default Routes

## Routing Packets: Default Gateway

- End hosts like PC1 and PC4 can send packets directly to destinations in their connected network.
    - PC1 is connected to 192.168.1.0/24, PC4 is connected to 192.168.4.0/24.
  - To send packets to destinations outside of their local network, they must send the packets to their **default gateway**.
 

**PC1 (Linux) Config:**

```
iface eth0 inet static
  address 192.168.1.10/24
  gateway 192.168.1.1
```

**PC4 (Linux) Config:**

```
iface eth0 inet static
  address 192.168.4.10/24
  gateway 192.168.4.4
```
  - The **default gateway** configuration is also called a **default route**.
    - It is a route to 0.0.0.0/0 = all netmask bits set to 0. Includes all addresses from 0.0.0.0 to 255.255.255.255.
  - End hosts usually have no need for any more specific routes.
    - They just need to know: to send packets outside of my local network, I should send them to my default gateway.
-

## Routing Packets: Static Routes

- When R1 receives frame from PC1, it will de-encapsulate (remove L2 header/trailer) and look at the inside packet
- It will check the routing table for the most specific matching route

```
C 192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
L   192.168.1.0/24 is directly connected, GigabitEthernet0/2
L   192.168.1.1/32 is directly connected, GigabitEthernet0/2
C 192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
L   192.168.12.0/24 is directly connected, GigabitEthernet0/1
L   192.168.12.1/32 is directly connected, GigabitEthernet0/1
C 192.168.13.0/24 is variably subnetted, 2 subnets, 2 masks
C   192.168.13.0/24 is directly connected, GigabitEthernet0/0
L   192.168.13.1/32 is directly connected, GigabitEthernet0/0
```

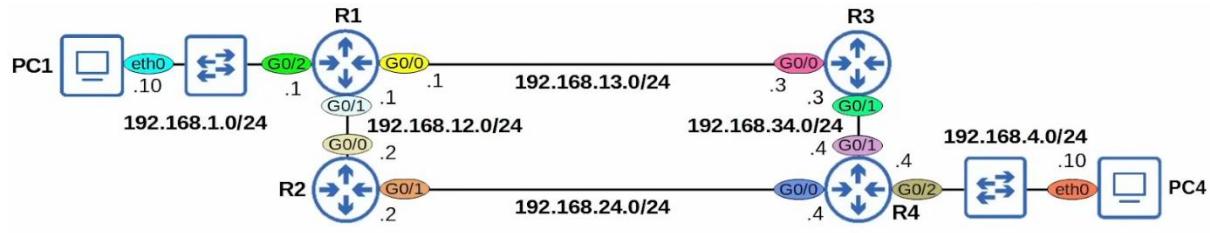
- R1 has no matching routes in its routing table
  - Drop the packet
- 2 possible paths
  - PC1->R1->R3->R4->PC2
  - PC1->R1->R2->R4->PC2
- Possible to configure the router to use both paths
  - Use as load balance - some packets go path 1 some go path 2
  - Use path 1 as main and path 2 as back up

## Static Route Configuration

- Each router needs 2 routes
  - Route to PC1
  - Route to PC4
  - Ensures **2-way reachability**
- Side note
  - Routers don't need routes to all networks in the path to the destination
  - They just need to know who to send next if want to send to a particular host
- R1 already has a Connected route to 192.168.1.0/24 (R1's network)

- R4 already has a Connected route to 192.168.4.0/24 (R4's network)
- Other routes must be manually configured

Router	Destination	Next-Hop
R1	192.168.1.0/24	Connected
	192.168.4.0/24	192.168.13.3
R3	192.168.1.0/24	192.168.13.1
	192.168.4.0/24	192.168.34.4
R4	192.168.1.0/24	192.168.34.3
	192.168.4.0/24	Connected



```
R1(config)# ip route 192.168.4.0 255.255.255.0 192.168.13.3   R1(config)# ip route ip-address netmask next-hop
R1(config)# do show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
!some code output omitted
Gateway of last resort is not set
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
        192.168.1.0/24 is directly connected, GigabitEthernet0/2
        L 192.168.1.1/32 is directly connected, GigabitEthernet0/2
      S 192.168.4.0/24 [1/0] via 192.168.13.3
      192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
        192.168.12.0/24 is directly connected, GigabitEthernet0/1
        L 192.168.12.1/32 is directly connected, GigabitEthernet0/1
      C 192.168.13.0/24 is variably subnetted, 2 subnets, 2 masks
        192.168.13.0/24 is directly connected, GigabitEthernet0/0
        L 192.168.13.1/32 is directly connected, GigabitEthernet0/0
The [1/0] displayed in static routes means:
[Administrative Distance/Metric]
We will cover these concepts later in the course.
```

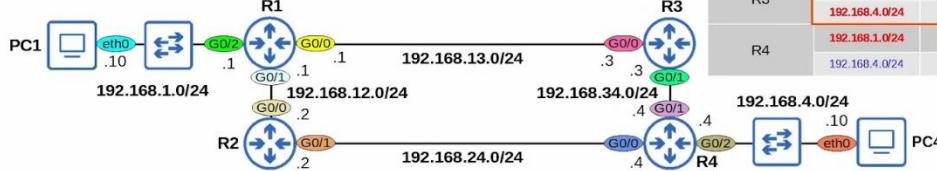
Router	Destination	Next-Hop
R1	192.168.1.0/24	Connected
	192.168.4.0/24	192.168.13.3
R3	192.168.1.0/24	192.168.13.1
	192.168.4.0/24	192.168.34.4
R4	192.168.1.0/24	192.168.34.3
	192.168.4.0/24	Connected

```
R3(config)# ip route 192.168.1.0 255.255.255.0 192.168.13.1      R3(config)# ip route ip-address netmask next-hop
```

```
R3(config)# do show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
!some code output omitted
```

Gateway of last resort is not set

```
S   192.168.1.0/24 [1/0] via 192.168.13.1
S   192.168.4.0/24 [1/0] via 192.168.34.4
      192.168.13.0/24 is variably subnetted, 2 subnets, 2 masks
C     192.168.13.0/24 is directly connected, GigabitEthernet0/0
L     192.168.13.3/32 is directly connected, GigabitEthernet0/1
C     192.168.34.0/24 is variably subnetted, 2 subnets, 2 masks
L     192.168.34.3/32 is directly connected, GigabitEthernet0/1
```



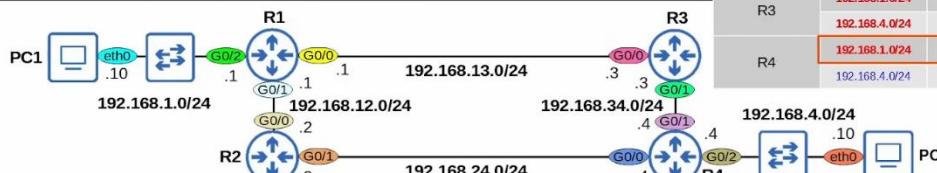
Router	Destination	Next-Hop
R1	192.168.1.0/24	Connected
	192.168.4.0/24	192.168.13.3
R3	192.168.1.0/24	192.168.13.1
	192.168.4.0/24	192.168.34.4
R4	192.168.1.0/24	192.168.34.3
	192.168.4.0/24	Connected

```
R4(config)# ip route 192.168.1.0 255.255.255.0 192.168.34.3      R4(config)# ip route ip-address netmask next-hop
```

```
R4(config)# do show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
!some code output omitted
```

Gateway of last resort is not set

```
S   192.168.1.0/24 [1/0] via 192.168.34.3
      192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks
C     192.168.4.0/24 is directly connected, GigabitEthernet0/2
L     192.168.4.4/32 is directly connected, GigabitEthernet0/2
      192.168.24.0/24 is variably subnetted, 2 subnets, 2 masks
C     192.168.24.0/24 is directly connected, GigabitEthernet0/0
L     192.168.24.4/32 is directly connected, GigabitEthernet0/0
      192.168.34.0/24 is variably subnetted, 2 subnets, 2 masks
C     192.168.34.0/24 is directly connected, GigabitEthernet0/1
L     192.168.34.4/32 is directly connected, GigabitEthernet0/1
```



Router	Destination	Next-Hop
R1	192.168.1.0/24	Connected
	192.168.4.0/24	192.168.13.3
R3	192.168.1.0/24	192.168.13.1
	192.168.4.0/24	192.168.34.4
R4	192.168.1.0/24	192.168.34.3
	192.168.4.0/24	Connected

```

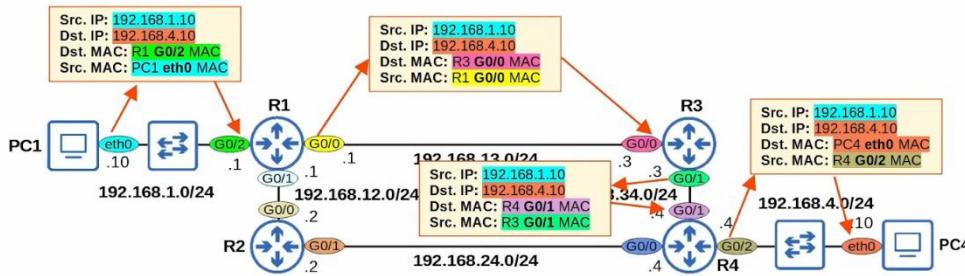
PC1:~$ ping 192.168.4.10
PING 192.168.4.10 (192.168.4.10): 56 data bytes
64 bytes from 192.168.4.10: seq=0 ttl=42 time=8.745 ms
64 bytes from 192.168.4.10: seq=1 ttl=42 time=4.423 ms
64 bytes from 192.168.4.10: seq=2 ttl=42 time=3.428 ms
64 bytes from 192.168.4.10: seq=3 ttl=42 time=3.544 ms
64 bytes from 192.168.4.10: seq=4 ttl=42 time=3.520 ms
^C
--- 192.168.4.10 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 3.428/4.732/8.745 ms

```

If the ping is successful, that means there is two-way reachability.  
PC1 can reach PC4, and PC4 can reach PC1.

#### Packet traveling from PC1 to PC4:

\*we will examine this step-by-step in the "Life of a Packet" video



- Every time the packet is encapsulated and de-encapsulated, the source and destination MAC address changes while the source and destination UP remains the same

```

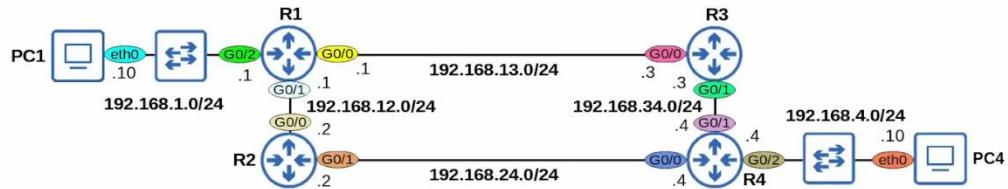
R2(config)# ip route 192.168.1.0 255.255.255.0 g0/0
R2(config)# ip route 192.168.4.0 255.255.255.0 g0/1 192.168.24.4

R2(config)# do show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
!some code output omitted
Gateway of last resort is not set
      
```

```

R2(config)# ip route ip-address netmask exit-interface
R2(config)# ip route ip-address netmask exit-interface next-hop
      
```

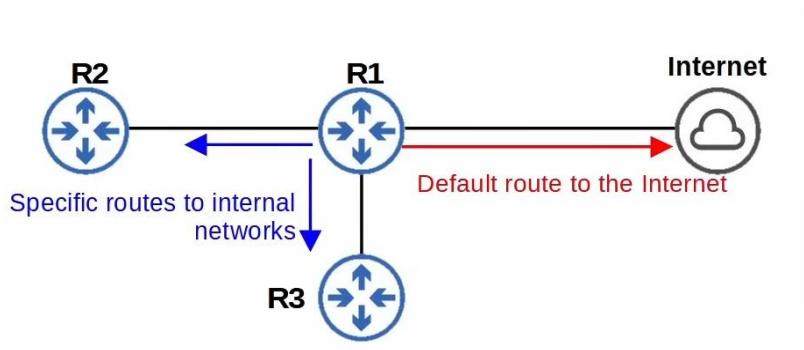
- Static routes in which you specify only the `exit-interface` rely on a feature called **Proxy ARP** to function.
- This is usually not a problem, but generally you can stick to `next-hop` or `exit-interface next-hop`.
- Neither is 'better' than the other: use which you prefer.



#### Default Route

- A **default route** is a route to 0.0.0.0/0
  - It is the least specific route possible, consists of all IP address
- If router don't have any more specific routes that match a packet's destination IP address, the router will forward the packet using the **default route**

- A default route is often used to direct traffic to the Internet
  - More specific routes are used for destinations in the internal corporate network
  - Traffic to destinations outside of the internal network is sent to the Internet



```
R1# show ip route
!codes omitted
Gateway of last resort is not set
No default route has been configured yet.

S      10.0.0.0/8 [1/0] via 192.168.12.2
S      172.16.0.0/16 [1/0] via 192.168.13.3
C      192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
L          192.168.12.1/32 is directly connected, GigabitEthernet0/1
C      192.168.13.0/24 is variably subnetted, 2 subnets, 2 masks
L          192.168.13.1/32 is directly connected, GigabitEthernet0/0
C      203.0.113.0/24 is variably subnetted, 2 subnets, 2 masks
L          203.0.113.0/24 is directly connected, GigabitEthernet0/2
C      203.0.113.1/32 is directly connected, GigabitEthernet0/2
L          203.0.113.1/32 is directly connected, GigabitEthernet0/2
```



```
R1(config)# ip route 0.0.0.0 0.0.0.0 203.0.113.2
R1(config)# do show ip route
!most codes omitted
    ia - IS-IS inter area, [* - candidate default], U - per-user static route
!most codes omitted
Gateway of last resort is 203.0.113.2 to network 0.0.0.0

S*      0.0.0.0/0 [1/0] via 203.0.113.2
S      10.0.0.0/8 [1/0] via 192.168.12.2
S      172.16.0.0/16 [1/0] via 192.168.13.3
C          192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
L              192.168.12.1/32 is directly connected, GigabitEthernet0/1
C          192.168.13.0/24 is variably subnetted, 2 subnets, 2 masks
L              192.168.13.1/32 is directly connected, GigabitEthernet0/0
C          203.0.113.0/24 is variably subnetted, 2 subnets, 2 masks
L              203.0.113.0/24 is directly connected, GigabitEthernet0/2
C          203.0.113.1/32 is directly connected, GigabitEthernet0/2
L              203.0.113.1/32 is directly connected, GigabitEthernet0/2
```

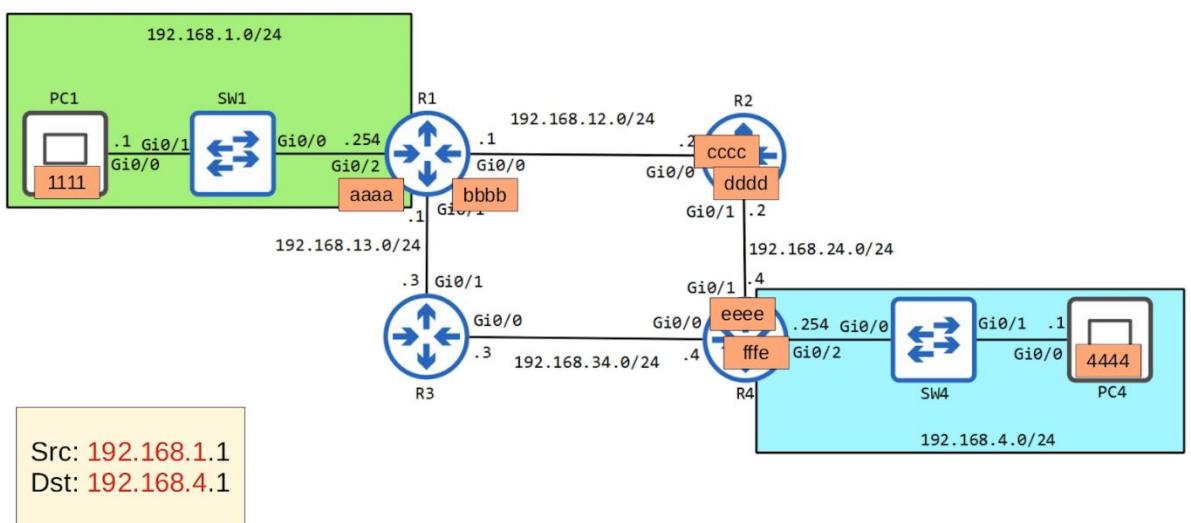
## Things Covered

- Intro to Static Routes
- Static Route config
  - "ip route <ip-address> <netmask> <next-hop>"
  - "ip route <ip-address> <netmask> <exit-interface>"
  - "ip route <ip-address> <netmask> <exit-interface next-hop>"
- Default routes

## Life of a Packet

### Things Covered

- Entire process of sending a packet to a remote destination
  - Includes ARP, encapsulation and decapsulation



### PC1 -> R1

- Sending from PC1 to PC4

- PC1 realises that PC4 is not in the same network due to different network address
- PC1 will need to send to the default gateway
- PC1 need to send ARP request (broadcast) to find the MAC address of R1

**ARP Request**

Src IP: 192.168.1.1  
 Dst IP: 192.168.1.254  
 Dst MAC: ffff.ffff.ffff  
 Src MAC: 1111

**ARP Reply**

Src IP: 192.168.1.254  
 Dst IP: 192.168.1.1  
 Dst MAC: 1111  
 Src MAC: aaaa

- R1 sends an ARP reply (unicast) to PC1
- Once PC1 has the MAC address, send to R1

### R1->R2

- R1 receives and removes the Ethernet header
- R1 looks up the destination in its own routing table

R1 Routing Table	
Destination	Next Hop
192.168.4.0/24	192.168.12.2
...	

- R1 now needs to encapsulate the packet with a new Ethernet header with the MAC address of 192.168.12.2
- R1 However, don't know the MAC address
- R1 Send ARP request
- R2 send ARP reply
- R1 now knows R2 MAC address and send the packet to R2

## R2->R4

- R2 removes Ethernet header
- Looks up destination on its own routing table

R2 Routing Table	
Destination	Next Hop
192.168.4.0/24	192.168.24.4
...	..

- Same process as R1->R2
  - Send ARP request
  - Get MAC address
  - Send to R4

## R4->PC4

- Same thing as above
- Send ARP since don't know MAC of PC4
- IP header remained the same throughout
- Switches don't encapsulate/de-encapsulate the packet
  - They only learn the MAC address of device when packets are sent on the interface

## PC4->PC1 (Reversed)

- Now all the routers have the MAC address of the other routers in the path
- No ARP request are sent

# Subnetting

## Things covered

- CIDR (Classless Inter-Domain Routing)

- Process of subnetting

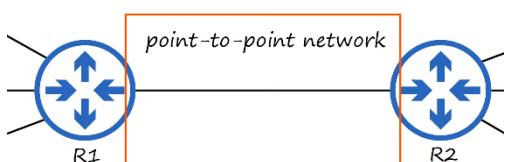
## IPv4 Address Classes

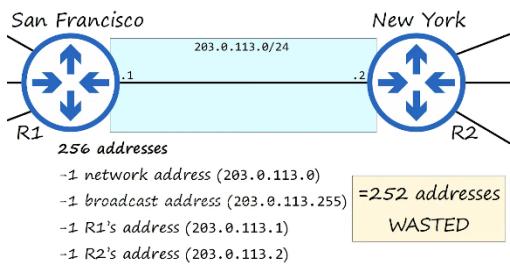
Class	First octet (binary)	First octet range (decimal)	
A	0xxxxxxx	0 - 127	0.0.0.0 ~ 127.255.255.255
B	10xxxxxx	128 - 191	128.0.0.0 ~ 191.255.255.255
C	110xxxxx	192 - 223	192.0.0.0 ~ 223.255.255.255
D	1110xxxx	224 - 239	224.0.0.0 ~ 239.255.255.255
E	1111xxxx	240 - 255	240.0.0.0 ~ 255.255.255.255

Class	First octet	First octet numeric range	Prefix Length
A	0xxxxxxx	0-127	/8
B	10xxxxxx	128-191	/16
C	110xxxxx	192-223	/24

Class	Leading bits	Size of network number bit field	Size of rest bit field	Number of networks	Addresses per network
Class A	0	8	24	128 ( $2^7$ )	16,777,216 ( $2^{24}$ )
Class B	10	16	16	16,384 ( $2^{14}$ )	65,536 ( $2^{16}$ )
Class C	110	24	8	2,097,152 ( $2^{21}$ )	256 ( $2^8$ )

- The IANA (Internet Assigned Numbers Authority) assigns IPv4 addresses/networks to companies based on their size
- E.g. A large company may receive class A network while a small company receives a class C network
- However, leads to many wasted IP addresses





## Classless Inter-Domain Routing (CIDR)

- Requirements for Class A = /8, Class B = /16, Class C = /24 were removed
- Allows larger networks to be split into smaller networks, allowing greater efficiency
- These smaller networks are called 'subnetworks' or 'subnets'
- Can have
  - /25 - 255.255.255.128 ->  $2^7 - 2 = 126$  usable address
  - /26 - 255.255.255.192 ->  $2^6 - 2 = 62$  usable address
  - /27 - 255.255.255.224 ->  $2^5 - 2 = 30$  usable address
  - /28 - 255.255.255.240 ->  $2^4 - 2 = 14$  usable address
  - /29 - 255.255.255.248 ->  $2^3 - 2 = 6$  usable address
  - /30 - 255.255.255.252 ->  $2^2 - 2 = 2$  usable address
    - E.g. If use /30 the remaining address blocks in /24 can still be used in other subnets
  - /31 - 255.255.255.254 ->  $2^1 - 2 = 0$  usable address
    - Can be used in point-to-point networks
    - No need for network address and broadcast address
    - Used for the 2 routers

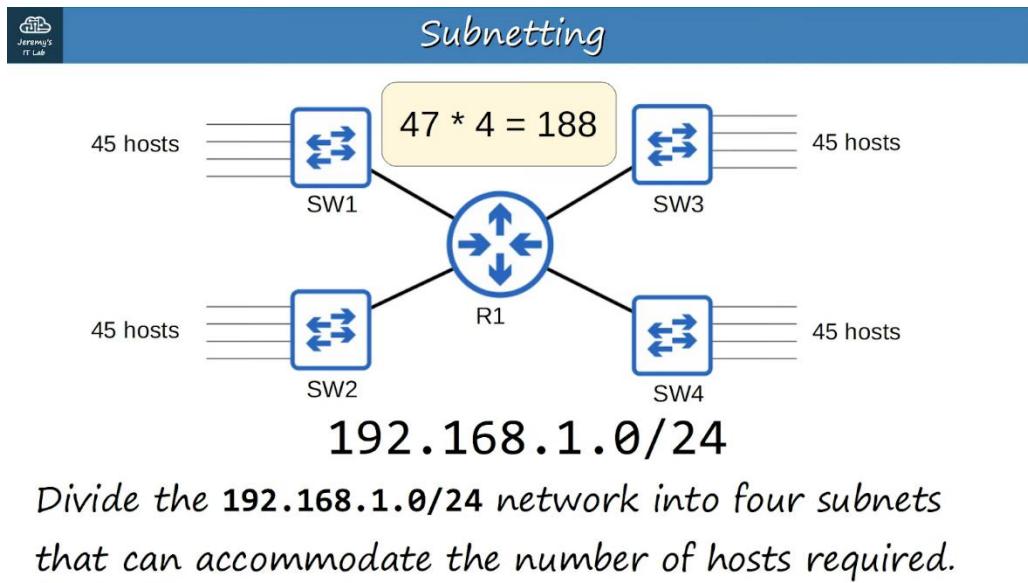
```

 203.0.113.0/31
 = 203.0.113.0 - 203.0.113.1
 11001011.00000000.01110001.00000000
Router(config-if)#ip address 203.0.113.0 255.255.255.254
% Warning: use /31 mask on non point-to-point interface cautiously
Router(config-if)#

```

- /32 - 255.255.255.255
  - Can be used to create a static route to 1 specific host

Dotted Decimal	CIDR Notation
255.255.255.128	/25
255.255.255.192	/26
255.255.255.224	/27
255.255.255.240	/28
255.255.255.248	/29
255.255.255.252	/30
255.255.255.254	/31
255.255.255.255	/32

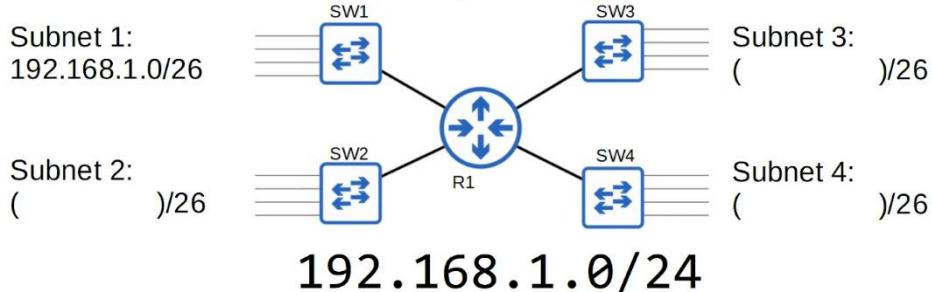




## QUIZ

The first subnet (Subnet 1) is 192.168.1.0/26. What are the remaining subnets?

HINT: Find the broadcast address of Subnet 1. The next address is the network address of Subnet 2. Repeat the process for Subnets 3 and 4.



$2^6 = 64$  for each subnet

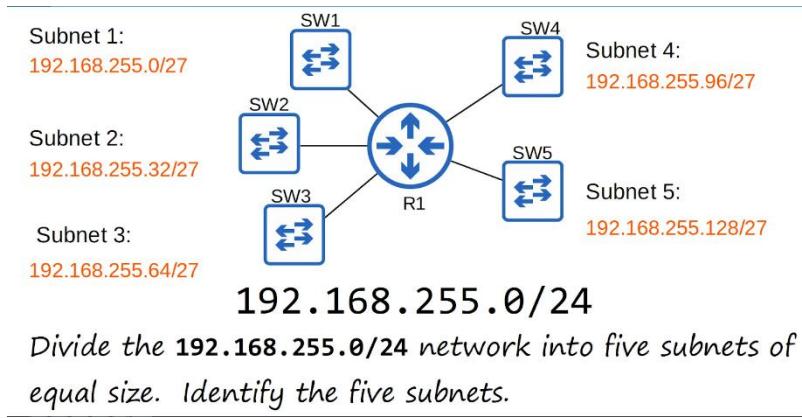
Subnet 1: 0 - 63

Subnet 2: 64 - 127

Subnet 3: 128 - 191

Subnet 4: 192 - 255

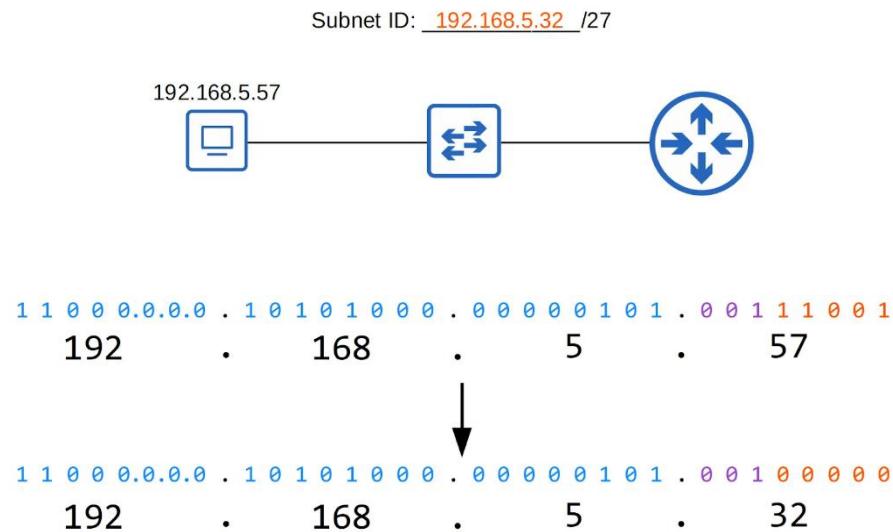
Example



- When given n networks to divide
  - $2^b \geq n$
  - E.g.  $2^3 = 8 \leq 5$
  - Last octet = **1110 0000**
  - Network portion of last octet =  $111 - 128+64+32$
  - Address of each subnet is 0, 0+32, 0+32+32...

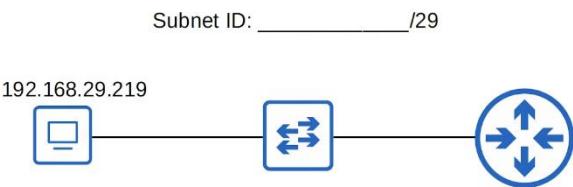
### Example

What subnet does host **192.168.5.57/27** belong to?



### Example

What subnet does host **192.168.29.219/29** belong to?



- Last octet = 1101 1011
- /29 -> 5 MSB
- Network portion = 1101 1000 -> 216
- Answer = 192.168.29.216

### Subnets/Hosts (Class C)

Prefix Length	Number of Subnets	Number of Hosts
/25	2	126
/26	4	62
/27	8	30
/28	16	14
/29	32	6
/30	64	2
/31	128	0 (2)
/32	256	0 (1)

- For /31, can be used for point-to-point connection
  - With no network and broadcast address
- For /32, can be used for routing

### Class B Subnetting

- Exactly same process as Class C

### Example

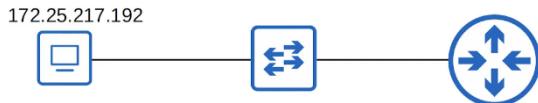
You have been given the 172.16.0.0/16 network. You are asked to create 80 subnets for your company's various LANs. What prefix length should you use?

## 172.16.0.0/16

- $2^6 = 64$
- $2^7 = 128$
- 7 bits needed for last 2 octets for network portion
- 1111 1110. 0000 0000
- Subnet 1 = 172.16.0.0
- Subnet 2 = 172.16.2.0
- Multiples of 2
- Therefore  $16+7 = /23$

What subnet does host **172.25.217.192/21** belong to?

Subnet ID: \_\_\_\_\_/21



- $217 \rightarrow 1101\ 1001$
- $1101\ 1000 = 216$
- Ans = 172.25.216.0

Prefix Length	Number of Subnets	Number of Hosts
/17	2	32766
/18	4	16382
/19	8	8190
/20	16	4094
/21	32	2044
/22	64	1022
/23	128	510
/24	256	254

Prefix Length	Number of Subnets	Number of Hosts
/25	512	126
/26	1024	62
/27	2048	30
/28	4096	14
/29	8192	6
/30	16384	2
/31	32768	0 (2)
/32	65536	0 (1)

## Class A Subnetting

You have been given the 10.0.0.0/8 network. You must create 2000 subnets which will be distributed to various enterprises.

What prefix length must you use?

How many host addresses (usable addresses) will be in each subnet?

- $2^{10} = 1,024$
- $2^{11} = 2,048$
- Ans :  $8 + 11 = /19$
- Num of host address bits =  $24 - 11 = 13$
- $2^{13} = 8,192 - 2 = 8,190$

PC1 has an IP address of **10.217.182.223/11**.

Identify the following for PC1's subnet:

- 1) Network address:
- 2) Broadcast address:
- 3) First usable address:
- 4) Last usable address:
- 5) Number of host (usable) addresses:



Q1)

- $217 \rightarrow \underline{1101} \ 1001$

- 1100 0000 -> 192
- 10.192.0.0

Q2)

- 1101 1111 -> 223
- 10.223.255.255

Q3)

- 10.192.0.1

Q4)

- 10.223.255.254

Q5)

- $2^{(32-11)} = 2,097,152$
- $2097152 - 2 = 2\ 097\ 150$

### Variable Length Subnet Masks (VLSM)

- VLSM is the process of creating subnets of different sizes, to make your use of network addresses more efficient
- More complicated than FLSM

### Step to solve VLSM

1. Start from the largest to smallest
2. Assign the appropriate netmask size
3. Get the network and broadcast address
4. Use the next address after broadcast address for the next subnet
5. Repeat Step 1

# VLAN

## VLAN Intro

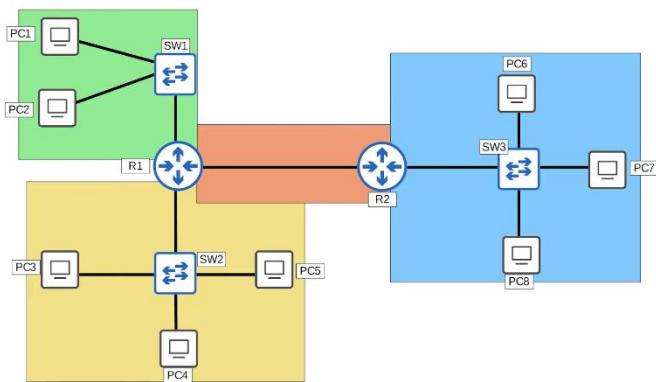
### Things Covered

- What is a LAN
- Broadcast domains
- What is a VLAN
- What is the purpose of VLAN
- How to configure VLAN

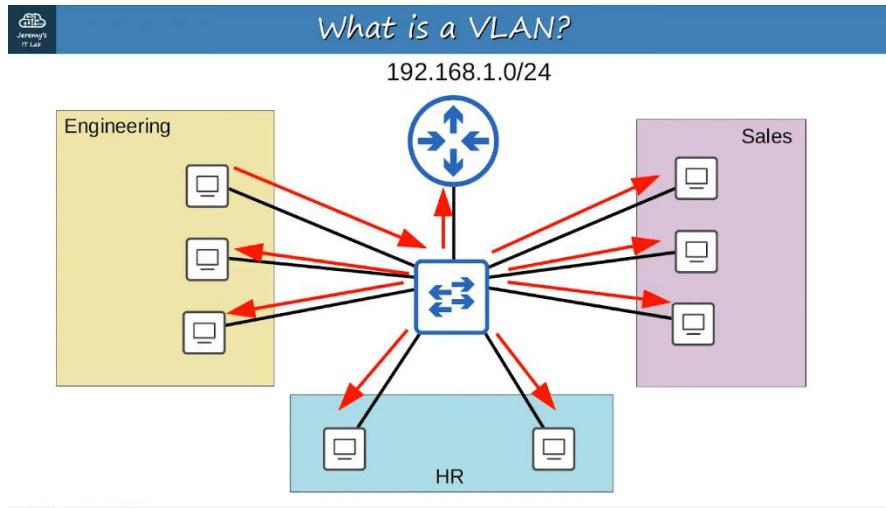
### What is a LAN

- LAN
  - A single **broadcast domain** including all devices in that broadcast domain
- A broadcast domain
  - The group of devices which will receive a broadcast frame (destination MAC FFFF.FFFF.FFFF) sent by any 1 of the members

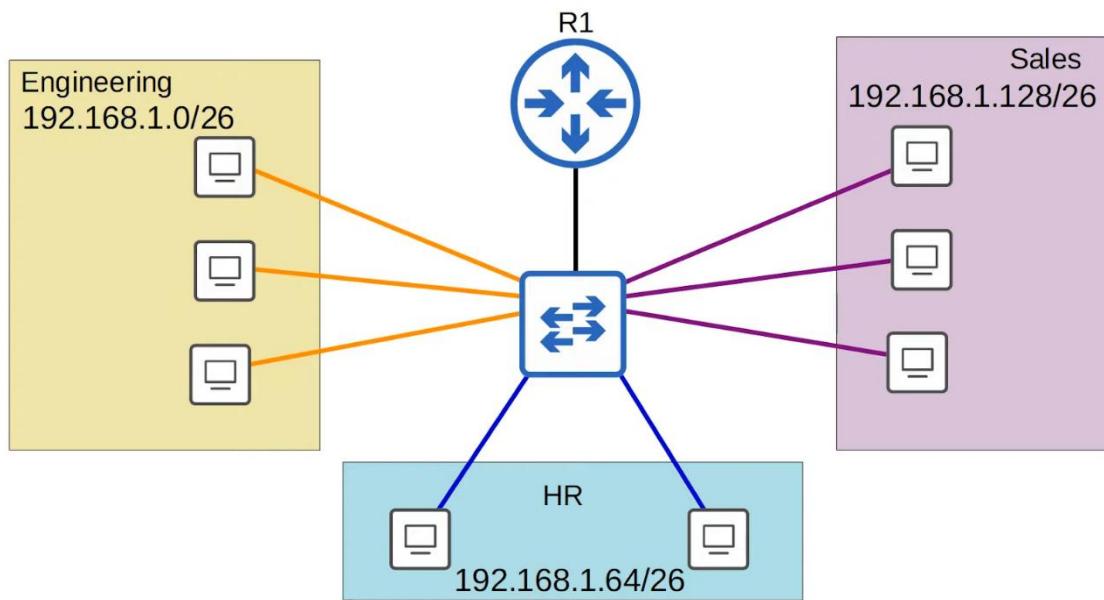
### Broadcast Domain/LANs



## What is a VLAN

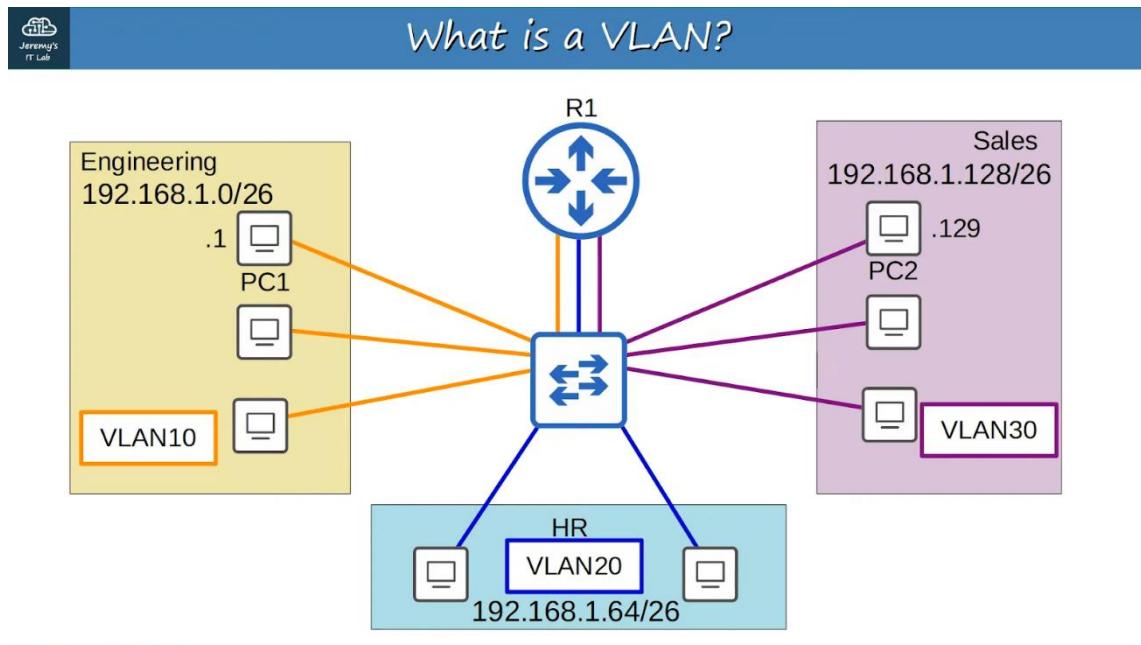


- Performance
  - Lots of unnecessary broadcast traffic can reduce network performance
- Security
  - Even within the same office, you want to limit who has access to what.  
You can apply security policies on a router/firewall
  - However, this is 1 LAN, so there won't be any effect



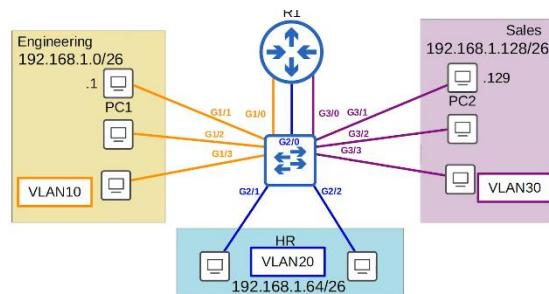
- When sending a broadcast message (MAC dest FFFF.FFFF.FFFF) from PC1, it will still be sent to all PCs even though they are in different subnets
- This is because switches operate at Layer 2 and only deal with MAC address

- Although we separated the 3 departments into 3 subnets (Layer 3) they are still in the same broadcast domain (Layer 2)



- Need to configure the switch
- Switch will not forward traffic btw VLANs, including broadcast/unknown unicast traffic
- Switch does not perform inter-VLAN routing, must send through the router
  - PC1 -> SW -> R1 -> SW -> PC2
- VLANs
  - Are configured on switches on a per-interface basis
  - Logically separate end hosts at Layer 2
- Switches do not forward traffic directly btw hosts in different VLANs

### Configuration



```

SW1#show vlan brief

VLAN Name          Status    Ports
-----              -----
1     default       active    Gi0/0, Gi0/1, Gi0/2, Gi0/3
                      Gi1/0, Gi1/1, Gi1/2, Gi1/3
                      Gi2/0, Gi2/1, Gi2/2, Gi2/3
                      Gi3/0, Gi3/1, Gi3/2, Gi3/3
1002 fddi-default   act/unsup
1003 token-ring-default   act/unsup
1004 fddinet-default   act/unsup
1005 trnet-default   act/unsup
SW1#

```

- VLANs 1, 1002-1005 exists by default, cannot be deleted
- 1002-1005 are old technologies, can ignore for CCNA

```

SW1(config)#interface range g1/0 - 3
SW1(config-if-range)#switchport mode access
SW1(config-if-range)#switchport access vlan 10
% Access VLAN does not exist. Creating vlan 10
SW1(config-if-range)#interface range g2/0 - 2
SW1(config-if-range)#switchport mode access
SW1(config-if-range)#switchport access vlan 20
% Access VLAN does not exist. Creating vlan 20
SW1(config-if-range)#interface range g3/0 - 3
SW1(config-if-range)#switchport mode access
SW1(config-if-range)#switchport access vlan 30
% Access VLAN does not exist. Creating vlan 30
SW1(config-if-range)#

```

- "switchport mode access" sets the interface as an access port
- Access port
  - A switchport that belongs to a single VLAN, and usually connects to end hosts
  - Switchports which carry multiple VLANs are called 'trunk ports'

```

SW1(config)#do show vlan brief

VLAN Name          Status    Ports
-----              -----
1     default       active    Gi0/0, Gi0/1, Gi0/2, Gi0/3
                      Gi2/3
10    VLAN0010      active    Gi1/0, Gi1/1, Gi1/2, Gi1/3
20    VLAN0020      active    Gi2/0, Gi2/1, Gi2/2
30    VLAN0030      active    Gi3/0, Gi3/1, Gi3/2, Gi3/3
1002 fddi-default   act/unsup
1003 token-ring-default   act/unsup
1004 fddinet-default   act/unsup
1005 trnet-default   act/unsup
SW1(config)#vlan 10
SW1(config-vlan)#name ENGINEERING
SW1(config-vlan)#vlan 20
SW1(config-vlan)#name HR
SW1(config-vlan)#vlan 30
SW1(config-vlan)#name SALES

```

- "vlan 10" can also be used to create VLAN

```

SW1(config)#do show vlan brief
VLAN Name          Status    Ports
----  -----
1    default        active   Gi0/0, Gi0/1, Gi0/2, Gi0/3
                                Gi2/3
10   ENGINEERING   active   Gi1/0, Gi1/1, Gi1/2, Gi1/3
20   HR             active   Gi2/0, Gi2/1, Gi2/2
30   SALES          active   Gi3/0, Gi3/1, Gi3/2, Gi3/3
1002  fddi-default act/unsup
1003  token-ring-default act/unsup
1004  fddinet-default act/unsup
1005  trnet-default  act/unsup
SW1(config)#

```

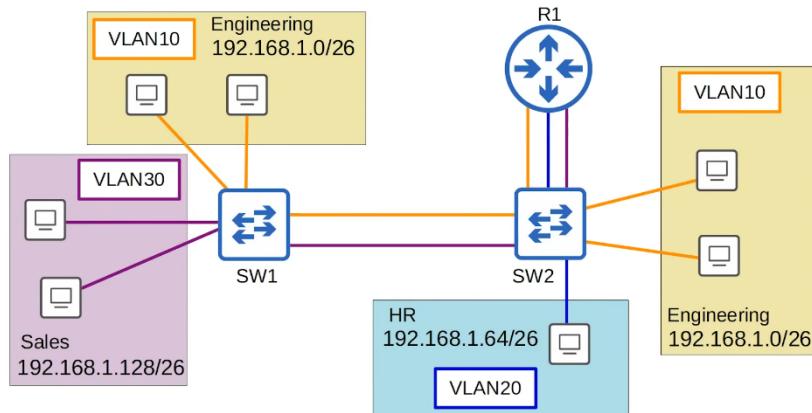
## Commands

- **"show vlan brief"**
- **"interface <interfaces>**
  - <interfaces> : "range f0/1-5"
- **"switchport mode access"**
- **"switchport access vlan <VLAN\_number>"**
  - OR from global config mode, NOT interface mode
  - **"vlan <VLAN\_number>"**

## Trunk Ports

### Things covered

- What is a trunk port
- What is the purpose of trunk ports
- 802.1Q encapsulation
- How to configure
- 'Router on a stick' (ROAS)



- There is no link in VLAN20 btw SW1 and SW2
- This is because there are no PCs in VLAN20 connected to SW1
- PCs in VLAN20 can still reach PCs connected to SW1, R1 will perform inter-VLAN routing

## Trunk Ports

- For a small network for a few VLANs, possible to use separate interface for each VLAN
- However, if there is a large number of VLANs, it will lead to wasted interfaces, and routers often don't have enough interfaces for each VLAN
- Can use trunk ports to carry traffic from multiple VLANs on single interface
  - Allows the receiving switch to know which VLAN the frame belongs to
- Trunk ports = Tagged ports
- Access ports = Untagged ports

## VLAN Tagging

- There are 2 main trunking protocols
  - ISL (Inter Switch Link)
  - 802.1Q (dot 1 q)

- ISL is old and not used in the real world
- Only need to know .1Q for CCNA



- The 802.1Q tag is inserted btw the Source and Type/Length field of the Ethernet frame
- 4 bytes
- Consists of 2 fields
  - Tag Protocol Identifier (TPI)
  - Tag Control Information (TCI)
- TCI consist of 3 sub fields

802.1Q tag format				
16 bits	3 bits	1 bit	12 bits	
TPID	TCI			
	PCP	DEI	VID	

- TPID (Tag Protocol Identifier)
  - 2 bytes (16 bits)
  - Always set to 0x8100. Indicates the frame is 802.1Q tagged
- PCP (Priority Code Point)
  - 3 bits
  - Used for Class of Service (CoS), which prioritizes important traffic in congested networks
- Drop Eligible Indicator (DEI)
  - 1 bit
  - Indicates if frame can be dropped if network is congested
- VID (VLAN ID)
  - 12 bits

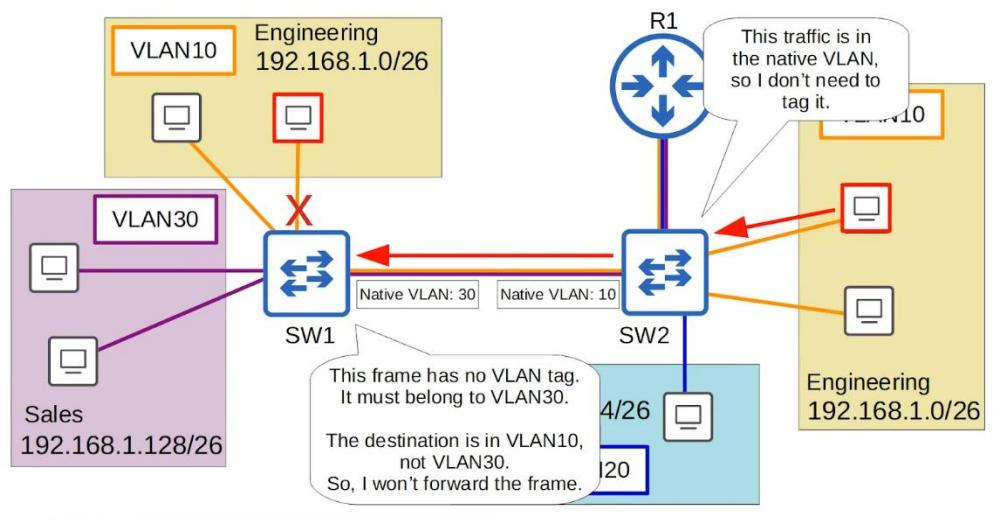
- Identifies the VLAN the frame belongs to
- 12 bits = 4096 VLANs, 0 - 4095
- However VLANs 0 and 4095 are reserved and cannot be used
- Actual range of VLANs: 1 - 4094
- Cisco's proprietary ISL also has a VLAN range 1 - 4094

## VLAN Ranges

- The range (1 - 4094) is divided into 2 sections
  - Normal VLANs : 1 - 1005
  - Extended VLANs : 1006 - 4094
- Some older devices don't support the extended VLANs, but modern switches do support them

## Native VLAN

- 802.1Q has a feature called the **native VLAN** (ISL don't have the feature)
- The native VLAN is VLAN1 by default on all trunk ports
  - However, can be manually configured on each trunk port
- The switch does not add an 802.1Q tag to frames in the native VLAN
- When a switch receives an untagged frame on a trunk port, it assumes the frame belongs to the native VLAN.
- **It's very important that the native VLAN matches**



## Configuration

```

SW1(config)#interface g0/0
SW1(config-if)#switchport mode trunk
Command rejected: An interface whose trunk encapsulation is "Auto" can not be configured to "trunk" mode.
SW1(config-if)#switchport trunk encapsulation ?
  dot1q      Interface uses only 802.1q trunking encapsulation when trunking
  isl       Interface uses only ISL trunking encapsulation when trunking
  negotiate  Device will negotiate trunking encapsulation with peer on
              interface

SW1(config-if)#switchport trunk encapsulation dot1q
SW1(config-if)#switchport mode trunk
SW1(config-if)#

```

- Many modern switches do not support Cisco's ISL at all. They only support dot1q
- However, switches that do both (like in the example) have a trunk encapsulation of 'Auto' by default
- To manually configure the interface as a trunk port, you must first set the encapsulation to dot1q or ISL
  - On switches that only have dot1q, not necessary
- After setting the encapsulation type, you can then configure the interface as a trunk

```

Sw1#show interfaces trunk
Port      Mode          Encapsulation  Status      Native vlan
Gi0/0    on           802.1q        trunking    1
Port      Vlans allowed on trunk
Gi0/0    1-4094
Port      Vlans allowed and active in management domain
Gi0/0    1,10,30
Port      Vlans in spanning tree forwarding state and not pruned
Gi0/0    1,10,30
Sw1#

```

```

Sw1#show vlan brief
VLAN Name                 Status Ports
1   default               active Gi1/1, Gi1/2, Gi1/3, Gi2/0
                           Gi2/1, Gi2/2, Gi2/3, Gi3/0
                           Gi3/1, Gi3/2, Gi3/3
10  ENGINEERING           active Gi0/1, Gi0/2
30  SALES                 active Gi0/3, Gi1/0
1002 fddi-default         act/unsup
1003 token-ring-default   act/unsup
1004 fddinet-default      act/unsup
1005 trnet-default        act/unsup
Sw1#

```

```

Sw1(config)#int g0/0
Sw1(config-if)#
Sw1(config-if)#switchport trunk allowed vlan ?
WORD    VLAN IDs of the allowed VLANs when this port is in trunking mode
add    add VLANs to the current list
all    all VLANs
except all VLANs except the following
none   no VLANs
remove remove VLANs from the current list
Sw1(config-if)#switchport trunk allowed vlan

```

```

Sw1(config-if)#switchport trunk allowed vlan 10,30
Sw1(config-if)#do show interfaces trunk
Port      Mode          Encapsulation  Status      Native vlan
Gi0/0    on           802.1q        trunking    1
Port      Vlans allowed on trunk
Gi0/0    10,30
Port      Vlans allowed and active in management domain
Gi0/0    10,30
Port      Vlans in spanning tree forwarding state and not pruned
Gi0/0    10,30
Sw1(config-if)#

```

```

SW1(config-if)#switchport trunk allowed vlan add 20
SW1(config-if)#do show interfaces trunk

Port      Mode          Encapsulation  Status      Native vlan
Gi0/0    on           802.1q        trunking     1

Port      Vlans allowed on trunk
Gi0/0    10,20,30

Port      Vlans allowed and active in management domain
Gi0/0    10,30

Port      Vlans in spanning tree forwarding state and not pruned
Gi0/0    10,30
SW1(config-if)#

```

- VLAN20 has not been created yet, so not in the "Vlans allowed and active in management domain"

```

SW1(config-if)#switchport trunk allowed vlan remove 20
SW1(config-if)#do show interfaces trunk

Port      Mode          Encapsulation  Status      Native vlan
Gi0/0    on           802.1q        trunking     1

Port      Vlans allowed on trunk
Gi0/0    10,30

Port      Vlans allowed and active in management domain
Gi0/0    10,30

Port      Vlans in spanning tree forwarding state and not pruned
Gi0/0    10,30
SW1(config-if)#

```

```

SW1(config-if)#switchport trunk allowed vlan all
SW1(config-if)#do show interfaces trunk

Port      Mode          Encapsulation  Status      Native vlan
Gi0/0    on           802.1q        trunking     1

Port      Vlans allowed on trunk
Gi0/0    1-4094

Port      Vlans allowed and active in management domain
Gi0/0    1,10,30

Port      Vlans in spanning tree forwarding state and not pruned
Gi0/0    1,10,30
SW1(config-if)#

```

```

SW1(config-if)#switchport trunk allowed vlan except 1-5,10
SW1(config-if)#do show interfaces trunk

Port      Mode          Encapsulation  Status      Native vlan
Gi0/0     on           802.1q        trunking    1

Port      Vlans allowed on trunk
Gi0/0     6-9,11-4094

Port      Vlans allowed and active in management domain
Gi0/0     30

Port      Vlans in spanning tree forwarding state and not pruned
Gi0/0     30
SW1(config-if)#

```

```

SW1(config-if)#switchport trunk allowed vlan none
SW1(config-if)#do show interfaces trunk

Port      Mode          Encapsulation  Status      Native vlan
Gi0/0     on           802.1q        trunking    1

Port      Vlans allowed on trunk
Gi0/0     none

Port      Vlans allowed and active in management domain
Gi0/0     none

Port      Vlans in spanning tree forwarding state and not pruned
Gi0/0     none
SW1(config-if)#

```

## VLAN

```

SW1(config-if)#switchport trunk allowed vlan 10,30
SW1(config-if)#do show interfaces trunk

Port      Mode          Encapsulation  Status      Native vlan
Gi0/0     on           802.1q        trunking    1

For security purposes, it is best to change the native VLAN to an unused VLAN.
(network security will be explained more in-depth later in the course)
**Make sure the native VLAN matches on between switches**

Port      Vlans allowed and active in management domain
Gi0/0     10,30

Port      Vlans in spanning tree forwarding state and not pruned
Gi0/0     10,30
SW1(config-if)#

```

```
SW1(config-if)#switchport trunk native vlan 1001
SW1(config-if)#do show interfaces trunk

Port      Mode          Encapsulation  Status        Native vlan
Gi0/0     on           802.1q         trunking    1001

Port      Vlans allowed on trunk
Gi0/0     10,30

Port      Vlans allowed and active in management domain
Gi0/0     10,30

Port      Vlans in spanning tree forwarding state and not pruned
Gi0/0     10,30
SW1(config-if)#[
```

---

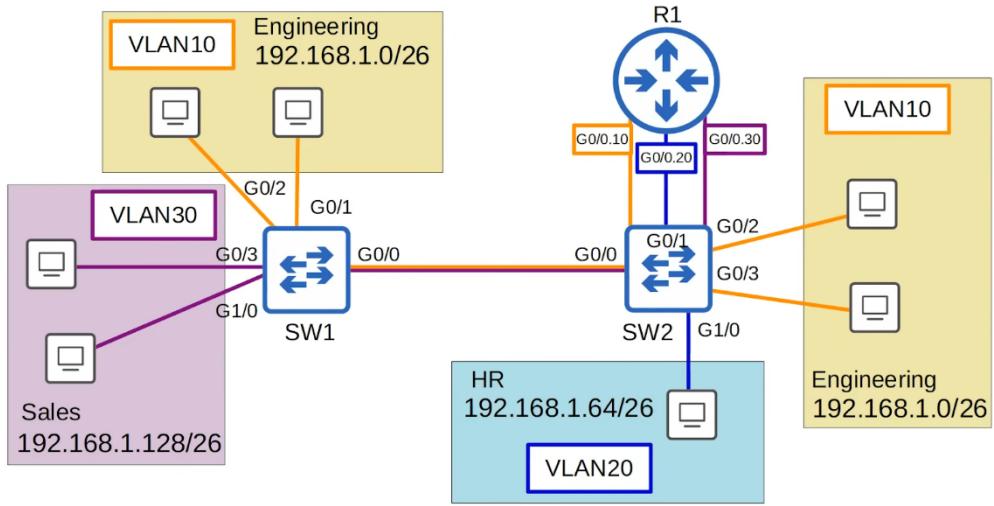
```
SW2(config)#interface g0/0
SW2(config-if)#switchport trunk encapsulation dot1q
SW2(config-if)#switchport mode trunk
SW2(config-if)#switchport trunk allowed vlan 10,30
SW2(config-if)#switchport trunk native vlan 1001
SW2(config-if)#do show interfaces trunk

Port      Mode          Encapsulation  Status        Native vlan
Gi0/0     on           802.1q         trunking    1001

Port      Vlans allowed on trunk
Gi0/0     10,30

Port      Vlans allowed and active in management domain
Gi0/0     10,30

Port      Vlans in spanning tree forwarding state and not pruned
Gi0/0     10,30
SW2(config-if)#[
```



```

R1(config)#interface g0/0
R1(config-if)#no shutdown
R1(config-if)#
*Apr 15 04:29:49.681: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up
*Apr 15 04:29:50.682: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up
R1(config-if)#interface g0/0.10
R1(config-subif)#encapsulation dot1q 10
R1(config-subif)#ip address 192.168.1.62 255.255.255.192
R1(config-subif)#interface g0/0.20
R1(config-subif)#encapsulation dot1q 20
R1(config-subif)#ip address 192.168.1.126 255.255.255.192
R1(config-subif)#interface g0/0.30
R1(config-subif)#encapsulation dot1q 30
R1(config-subif)#ip address 192.168.1.190 255.255.255.192
R1(config-subif)#

```

- Good practice to make the interface the same number as the VLAN
  - E.g. VLAN10 = g0/0.10

### ROAS (Router on a stick)

- Used to route btw multiple VLANs using a single interface on the router and switch
- The switch interface is configured as a regular trunk
- The router interface is configured using sub-interfaces
  - You configure the VLAN tag and IP address on each sub-interface
- The router will behave as if frames arriving with a certain VLAN tag have arrived on the sub-interface configured with that VLAN tag
- The router will tag frames sent out of each sub-interface with the VLAN tag configured on the sub-interface

## Commands

### Switches

- "**interface <interface>**"
- "**switchport trunk encapsulation dot1q**" OR  
**"switchport mode trunk"**
  - "**switchport trunk allowed vlan <VLAN\_number>**"
  - "switchport trunk allowed vlan add <VLAN\_number>"**
  - "switchport trunk allowed vlan all"**
  - "switchport trunk allowed vlan except <VLAN\_number>"**
  - "switchport trunk allowed vlan none"**
  - "switchport trunk allowed vlan remove <VLAN\_number>"**
    - <VLAN\_number>: can be "1-5,30"
    - "**show interfaces trunk**"
  - "**switchport trunk native vlan <VLAN\_number>**"
    - <VLAN\_number> should be an unused one for security purposes
    - E.g. 1001

### Routers

- "**interface <interface>**"
- "**no shutdown**"
- "**interface <sub-interface>**"
  - <sub-interface> : should be same as vlan\_number
  - E.g. g0/0.10
- "**encapsulation dot1q <VLAN\_number>"**
- "**ip address <ip\_address>"**

## SVI

Things covered

- Native VLAN on a route
- Wireshark analysis
- Layer 3 Switching/Multilayer Switching

Native VLAN

- 2 methods to configure native VLAN on a router
  - "encapsulation dot1q <vlan\_id> native"
    - Use on sub-interface
  - Configure the IP address for the native VLAN on the router's physical interface
    - Don't need to create a sub-interface, instead just key in ip address to the actual interface
    - E.g. set vlan10 as default, create sub-interface vlan20 and 30, the interface g0/0, set ip address to vlan10

Layer 3 (Multilayer) Switch



Layer 2 switch

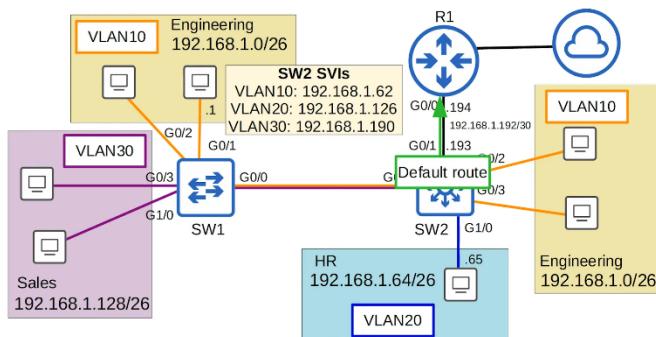
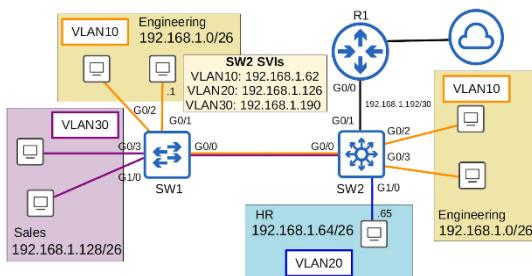


Layer 3 switch



- Capable of both switching and routing
- It is layer 3 aware
- Can assign IP addresses to the interfaces, like a router
- Can create virtual interfaces for each VLAN, and assign IP addresses to those interfaces
- Can configure routes, just like a router

- Can be used for inter-VLAN routing
- SVI (Switch Virtual Interface)
  - They are virtual interfaces you can assign IP addresses to in a multi-layer switch
  - Configure each PC to use the SVI (not the router) as their gateway address
  - To send traffic to different subnets/VLANs, the PCs will send traffic to the switch, and the switch will route the traffic



```
R1(config)#no interface g0/0.10
R1(config)#no interface g0/0.20
R1(config)#no interface g0/0.30
R1(config)#default interface g0/0
Interface GigabitEthernet0/0 set to default configuration
R1(config)#do show ip interface brief
Interface          IP-Address      OK? Method Status      Protocol
GigabitEthernet0/0  unassigned     YES NVRAM up        up
GigabitEthernet0/0.10  unassigned   YES manual deleted  down
GigabitEthernet0/0.20  unassigned   YES manual deleted  down
GigabitEthernet0/0.30  unassigned   YES manual deleted  down
GigabitEthernet0/1    unassigned     YES NVRAM administratively down  down
GigabitEthernet0/2    unassigned     YES NVRAM administratively down  down
```

```
R1(config)#interface g0/0
R1(config-if)#ip address 192.168.1.194 255.255.255.252
R1(config-if)#do show ip interface brief
Interface          IP-Address      OK? Method Status      Protocol
GigabitEthernet0/0  192.168.1.194  YES manual up        up
GigabitEthernet0/0.10 unassigned     YES manual deleted  down
GigabitEthernet0/0.20 unassigned     YES manual deleted  down
GigabitEthernet0/0.30 unassigned     YES manual deleted  down
GigabitEthernet0/1   unassigned     YES NVRAM administratively down down
GigabitEthernet0/2   unassigned     YES NVRAM administratively down down
```

```
SW2(config)#default interface g0/1
Interface GigabitEthernet0/1 set to default configuration
SW2(config)#ip routing
SW2(config)#interface g0/1
SW2(config-if)#no switchport
SW2(config-if)#ip address 192.168.1.193 255.255.255.252
SW2(config-if)#do show ip interface brief
Interface          IP-Address      OK? Method Status      Protocol
GigabitEthernet0/0  unassigned     YES unset up        up
GigabitEthernet0/2  unassigned     YES unset up        up
GigabitEthernet0/3  unassigned     YES unset up        up
GigabitEthernet0/1  192.168.1.193  YES manual up        up
GigabitEthernet1/0  unassigned     YES unset up        up
```

- "default interface g0/1"
  - Previously was used as trunk port
- "ip routing"
  - IMPORTANT
  - Allows the switch to act as multi-layer switch
- "no switchport"
  - Changes the switch from layer 2 to layer 3 switch

```
SW2(config-if)#exit
SW2(config)#ip route 0.0.0.0 0.0.0.0 192.168.1.194
SW2(config)#do show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
      a - application route
      + - replicated route, % - next hop override

Gateway of last resort is 192.168.1.194 to network 0.0.0.0

S*  0.0.0.0/0 [1/0] via 192.168.1.194
    192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C    192.168.1.192/30 is directly connected, GigabitEthernet0/1
L    192.168.1.193/32 is directly connected, GigabitEthernet0/1
SW2(config)#

```

SW2#show interfaces status							
Port	Name	Status	Vlan	Duplex	Speed	Type	
Gi0/0		connected	trunk	auto	auto	unknown	
Gi0/2		connected	10	auto	auto	unknown	
Gi0/3		connected	10	auto	auto	unknown	
Gi0/1		connected	routed	auto	auto	unknown	
Gi1/0		connected	20	auto	auto	unknown	
Gi1/1		connected	1	auto	auto	unknown	
Gi1/2		connected	1	auto	auto	unknown	
Gi1/3		connected	1	auto	auto	unknown	

## Configuring the SVIs

```
SW2(config)#interface vlan10
SW2(config-if)#ip address 192.168.1.62 255.255.255.192
SW2(config-if)#no shutdown
SW2(config-if)#interface vlan20
SW2(config-if)#ip address 192.168.1.126 255.255.255.192
SW2(config-if)#no shutdown
SW2(config-if)#interface vlan30
SW2(config-if)#ip address 192.168.1.190 255.255.255.192
SW2(config-if)#no shutdown
```

SW2(config-if)#do show ip interface brief						
Interface	IP-Address	OK?	Method	Status	Protocol	
GigabitEthernet0/0	unassigned	YES	unset	up	up	
GigabitEthernet0/2	unassigned	YES	unset	up	up	
GigabitEthernet0/3	unassigned	YES	unset	up	up	
GigabitEthernet0/1	192.168.1.193	YES	manual	up	up	
Vlan10	192.168.1.62	YES	manual	up	up	
Vlan20	192.168.1.126	YES	manual	up	up	
Vlan30	192.168.1.190	YES	manual	up	up	
Vlan40	40.40.40.40	YES	manual	down	down	

- For SVI to be in up/up state
  - The VLAN must exist on the switch
  - The switch must have at least 1 access port in the VLAN in an up/up state, AND/OR 1 trunk port that allows the VLAN that is an up/up state
  - The VLAN must not be shutdown
  - The SVI must not be shutdown (disabled by default)

## Commands

- Configure switch to layer 3 switch
  - "ip routing" (give permission to act as layer 3 switch)
  - "interface <interface>"
  - "no switchport" (enable layer 3 switch)
  - "interface <vlan num>"
  - "ip address <ip addr>"
  - "no shutdown"

## DTP & VTP

### Things covered

- Dynamic Trunking Protocol
- VLAN Trunking Protocol

### DTP (Dynamic Trunking Protocol)

- Is a Cisco proprietary protocol that allows Cisco switches to dynamically determine their interface status (access or trunk) w/o manual config
- Enabled by default on all Cisco switch interfaces
- So far, configure using
  - "switchport mode access" OR "switchport mode trunk"
  - With DTP, don't need to do so

- For security purposes, manual config is recommended
  - DTP should be disabled for all switchports

```
SW2(config-if)#switchport mode ?
access      Set trunking mode to ACCESS unconditionally
dot1q-tunnel  set trunking mode to TUNNEL unconditionally
dynamic     Set trunking mode to dynamically negotiate access or trunk mode
private-vlan Set private-vlan mode
trunk       Set trunking mode to TRUNK unconditionally
```

```
SW2(config-if)#switchport mode dynamic ?
auto        Set trunking mode dynamic negotiation parameter to AUTO
desirable   Set trunking mode dynamic negotiation parameter to DESIRABLE
```

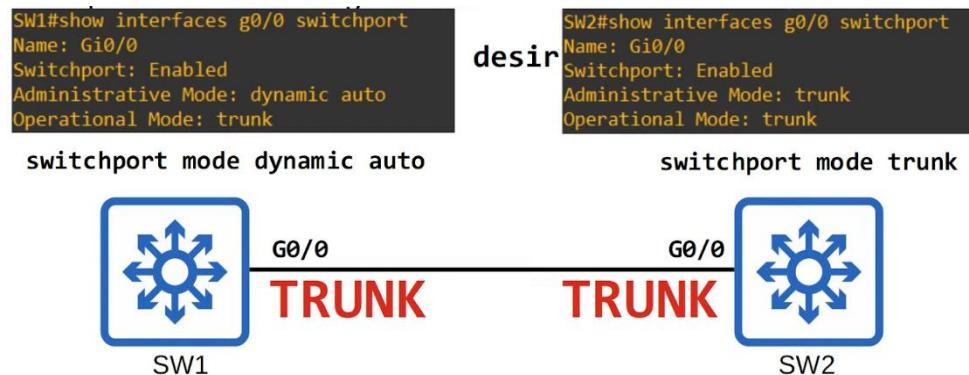
- "dynamic desirable"
  - Switchport will actively try to form a trunk with other Cisco switches
  - Will form a trunk if connected to another switchport in the following modes
    - "switchport mode trunk"
    - "switchport mode dynamic desirable"
    - "switchport mode dynamic auto"
- "dynamic auto"
  - Will not actively try to form a trunk with other Cisco switches
  - However, will form a trunk if other switch actively doing so
  - It will form a trunk in the following modes
    - "switchport mode trunk"
    - "switchport mode dynamic desirable"

## Dynamic Desirable

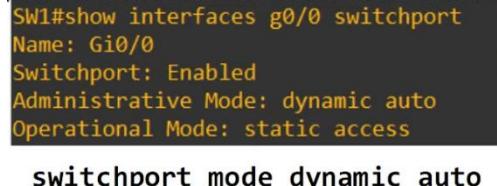


- "static access"
  - Access port that belongs to a single VLAN that doesn't change (unless you configure a different VLAN)
  - There are also "dynamic access" ports, in which a server auto assigns the VLAN depending on the MAC address of the connected device (not in CCNA)

## Dynamic Auto



~~SW1#show interfaces g0/0 switchport  
Name: Gi0/0  
Switchport: Enabled  
Administrative Mode: dynamic auto  
Operational Mode: static access~~



~~SW2#show interfaces g0/0 switchport  
Name: Gi0/0  
Switchport: Enabled  
Administrative Mode: dynamic auto  
Operational Mode: static access~~

switchport mode dynamic auto



```
SW1#show interfaces g0/0 switchport
Name: Gi0/0
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: static access
```

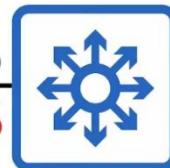
switchport mode dynamic auto



SW1

```
SW2#show interfaces g0/0 switchport
Name: Gi0/0
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
```

switchport mode access



SW2

c desira

Administrative Mode	Trunk	Dynamic Desirable	Access	Dynamic Auto
Trunk	Trunk	Trunk	X	Trunk
Dynamic Desirable	Trunk	Trunk	Access	Trunk
Access	X	Access	Access	Access
Dynamic Auto	Trunk	Trunk	Access	Access

- DTP will not form a trunk with a router, PC etc
- The switchport will be in access mode
- Older switches, "dynamic desirable" by default
- Newer switches, "dynamic auto" by default
- Can disable DTP on interface
  - "switchport nonegotiate"
- Configuring an access port with "switchport access mode" also disabled DTP on the interface
- Better to disable DTP for security

## Trunk Encapsulation

- Switches that support both 802.1q and ISL trunk encapsulations can use DTP to negotiate the encapsulation they will use
- This negotiation is enabled by default, as the default trunk encapsulation mode is
  - "switchport trunk encapsulation negotiate"
- ISL is favoured over 802.1q, so if both switches support ISL, it will be selected
- DTP frames are sent in VLAN1 when using ISL, or in the native VLAN when using 802.1q (the default native VLAN is VLAN1, however)

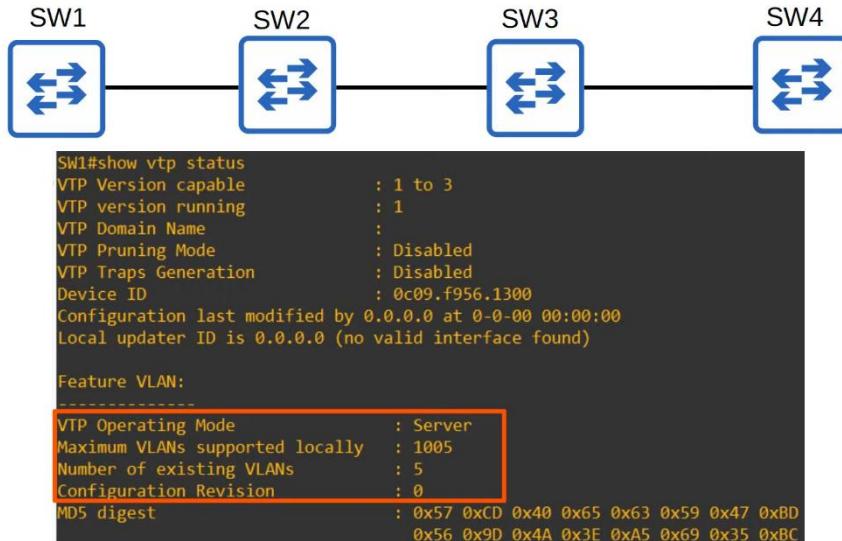
### VTP (VLAN Trunking Protocol)

- Allows you to config VLANs on a central VTP server switch, and other switches (VTP clients) will sync their VLAN database to the server
- It is designed for large networks with many VLANs, so that you don't have to config each VLAN on every switch
- It is rarely used, and recommended not to use
- 3 versions: 1,2,3
- 3 modes: server, client and transparent
- Cisco switches operate VTP server mode by default

### VTP Modes

- VTP Server
  - Can add/modify/delete VLANs
  - Store the VLAN DB in non-volatile RAM
  - Will increase the revision number every time a VLAN is added/modified/deleted
  - Will advertise the latest version of the VLAN DB on trunk interfaces, and the VTP clients will sync their VLAN DB to it
  - VTP servers also function as clients
    - VTP server will sync to another VTP server with a higher revision number
- VTP clients

- Cannot add/modify/delete
- Do not store VLAN DB in NVRAM (in VTP3, they do)
- Will sync their VLAN DB to the server with highest revision num in their VTP domain
- Will advertise their VLAN DB, and forward VTP advertisements to other clients over their trunk ports
- VLAN Transparent
  - Does not participate in the VTP domain (does not sync its VLAN DB)
  - Maintains its own VLAN DB in NVRAM
  - Can modify/add/delete VLANs, but won't advertise
  - Will forward VTP advertisements that are in the same domain
- To reset revision number
  - Change to transparent mode OR
  - Set domain name to an unused one



```

SW1(config)#vtp domain cisco
Changing VTP domain name from NULL to cisco
SW1(config)#
*May  4 02:14:47.276: %SW_VLAN-6-VTP_DOMAIN_NAME_CHG: VTP domain name changed to cisco.
SW1(config)#vlan 10
SW1(config-vlan)#name engineering
SW1(config-vlan)#exit

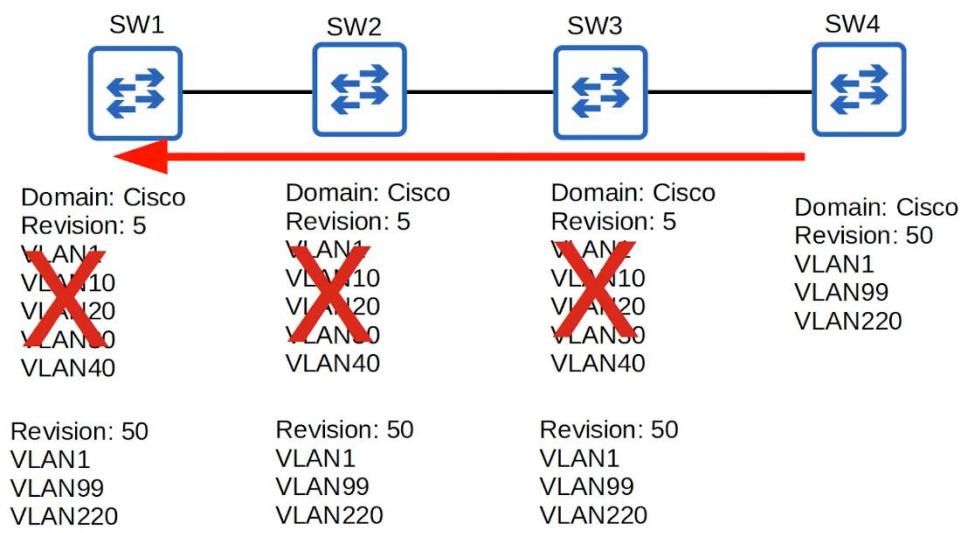
```

```

SW1#show vtp status
VTP Version capable      : 1 to 3
VTP version running     : 1
VTP Domain Name          : cisco
VTP Pruning Mode         : Disabled
VTP Traps Generation    : Disabled
Device ID                : 0c09.f956.1300
Configuration last modified by 0.0.0.0 at 5-4-20 02:18:27
Local updater ID is 0.0.0.0 (no valid interface found)

Feature VLAN:
-----
VTP Operating Mode       : Server
Maximum VLANs supported locally : 1005
Number of existing VLANs   : 6
Configuration Revision    : 1
MD5 digest               : 0x9F 0xE0 0xAB 0x7A 0x78 0x62 0x68 0x70
                           0x2D 0xD5 0x5A 0xBE 0x21 0x5D 0x56 0x49
SW1#

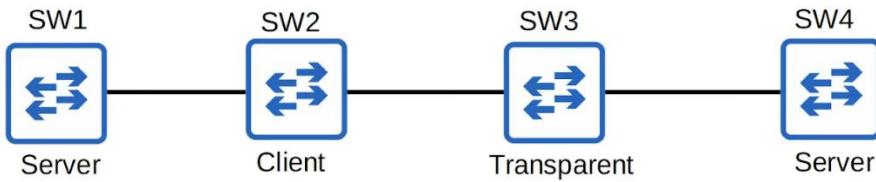
```



- If same name, dangerous



## VTP (VLAN Trunking Protocol)



```
SW1(config)#vlan 20
SW1(config-vlan)#name sales
SW1(config-vlan)#exit
SW1(config)#do show vlan brief

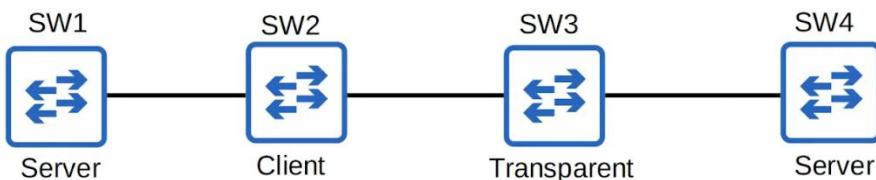
VLAN Name          Status   Po
----- 1 default      active   Gi
                                Gi
                                Gi
                                Gi
10 engineering    active
20 sales          active
1002 fddi-default act/unsup
1003 token-ring-default act/unsup
1004 fddinet-default act/unsup
1005 trnet-default act/unsup
SW1(config)#
```

```
SW1(config)#do show vtp status
VTP Version capable      : 1 to 3
VTP version running       : 1
VTP Domain Name           : cisco
VTP Pruning Mode          : Disabled
VTP Traps Generation      : Disabled
Device ID                 : 0c09.f956.1300
Configuration last modified by 0.0.0.0 at 5-4-20 03:40:01
Local updater ID is 0.0.0.0 (no valid interface found)

Feature VLAN:
-----
VTP Operating Mode        : Server
Maximum VLANs supported locally : 1005
Number of existing VLANs   : 7
Configuration Revision     : 4
MD5 digest                : 0x8F 0x9C 0x81 0x4B 0x5
                            0xE8 0xA3 0x98 0xFD 0xC
SW1(config)#
```



## VTP (VLAN Trunking Protocol)



```
SW2#show vlan brief

VLAN Name          Status   Po
----- 1 default      active   G
                                G
                                G
                                G
10 engineering    active
20 sales          active
1002 fddi-default act/unsup
1003 token-ring-default act/unsup
1004 fddinet-default act/unsup
1005 trnet-default act/unsup
SW2#
```

```
SW2#show vtp status
VTP Version capable      : 1 to 3
VTP version running       : 1
VTP Domain Name           : cisco
VTP Pruning Mode          : Disabled
VTP Traps Generation      : Disabled
Device ID                 : 0c09.f9ab.0800
Configuration last modified by 0.0.0.0 at 5-4-20 03:40:01

Feature VLAN:
-----
VTP Operating Mode        : Client
Maximum VLANs supported locally : 1005
Number of existing VLANs   : 7
Configuration Revision     : 4
MD5 digest                : 0x8F 0x9C 0x81 0x4B 0x5
                            0xE8 0xA3 0x98 0xFD 0xC
SW2#
```

 **VTP (VLAN Trunking Protocol)**



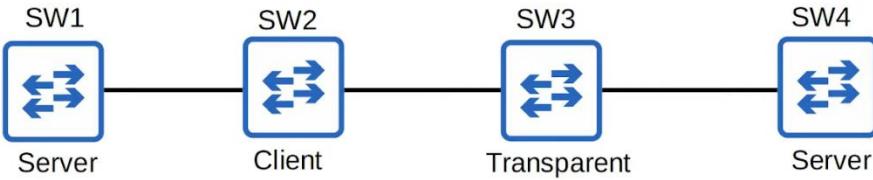
Changing the VTP domain to an unused domain will reset the revision number to 0.

```
SW3#show vlan brief
VLAN Name          Status    P
----- 
1 default          active    G
10 engineering     active    G
1002 fddi-default  act/unsup G
1003 token-ring-default  act/unsup G
1004 fddinet-default act/unsup G
1005 trnet-default  act/unsup G
SW3#
```

```
SW4#show vtp status
VTP Version capable      : 1 to 3
VTP version running       : 1
VTP Domain Name           : cisco
VTP Pruning Mode          : Disabled
VTP Traps Generation      : Disabled
Device ID                 : 0c09.f9fa.e700
Configuration last modified by 0.0.0 at 5-4-20 03:33:08
Local updater ID is 0.0.0 (no valid interface found)

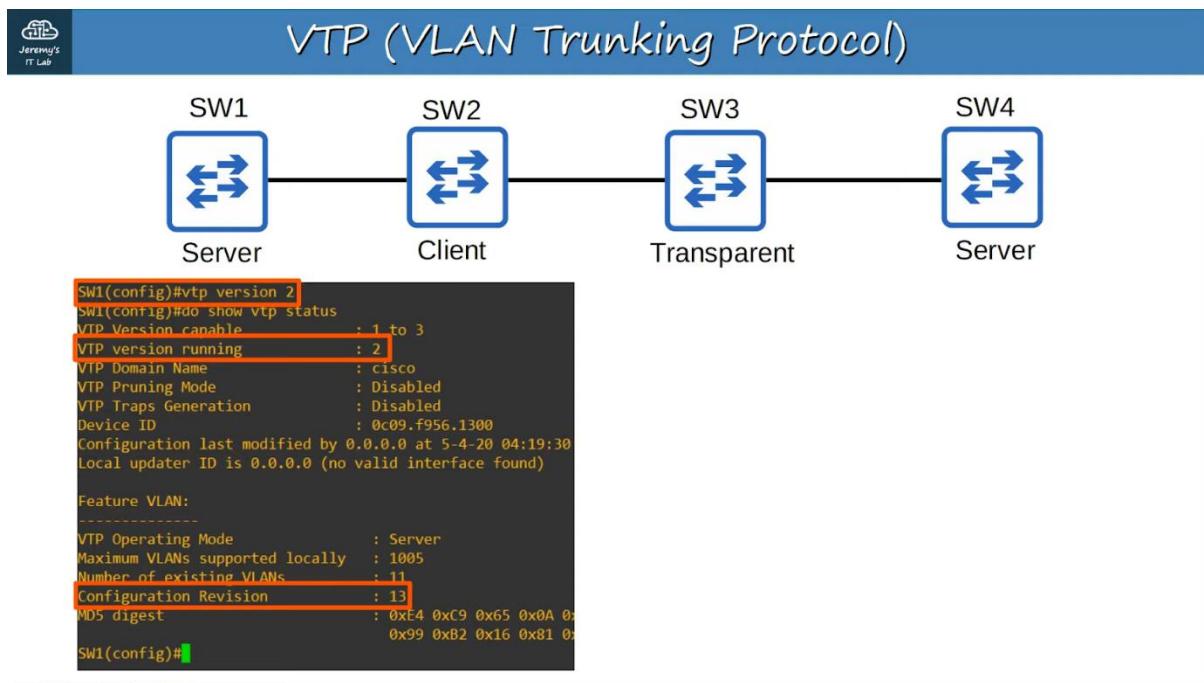
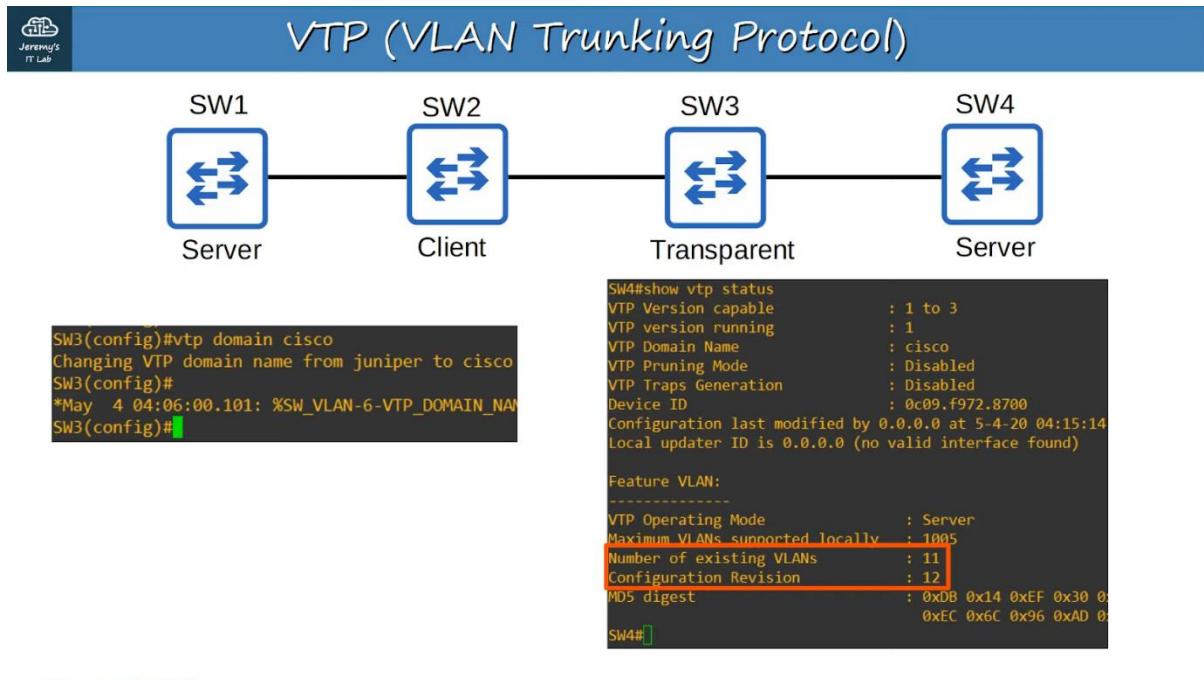
Feature VLAN:
-----
VTP Operating Mode        : Transparent
Maximum VLANs supported locally : 1005
Number of existing VLANs   : 6
Configuration Revision     : 0
MDS digest                : 0xDB 0x6A 0x82 0x61 0x
                            0x59 0x73 0x4E 0xF4 0x
SW4#
```

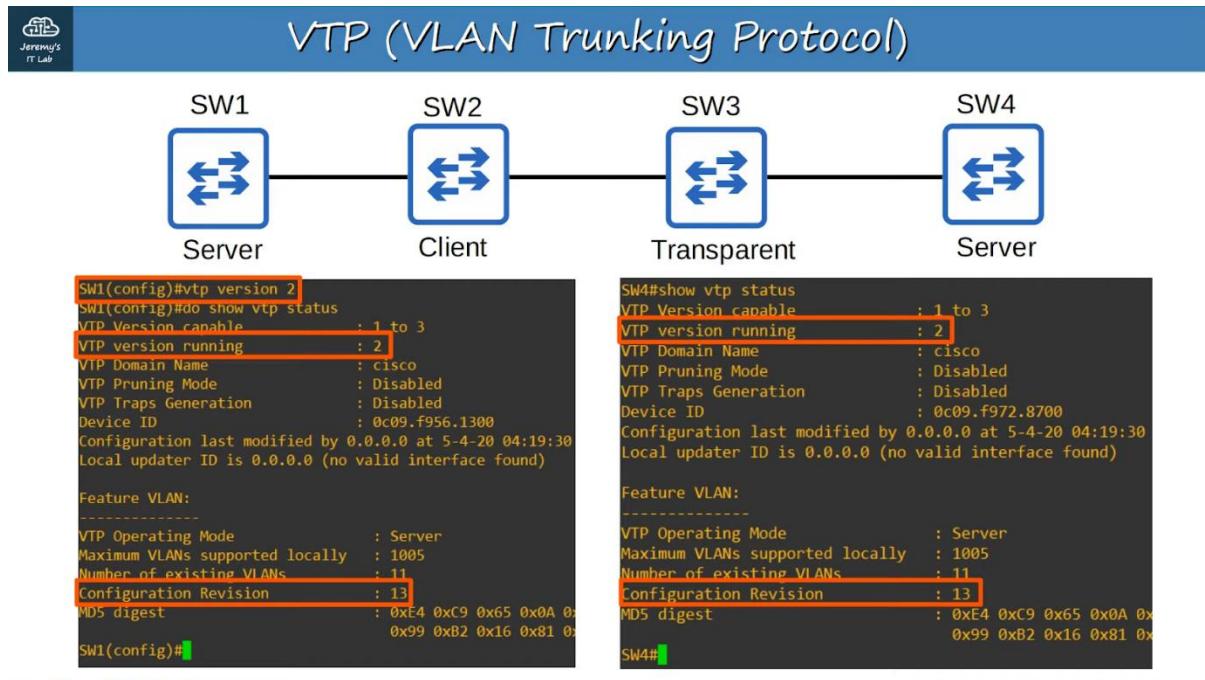
 **VTP (VLAN Trunking Protocol)**



```
SW4#show vtp status
VTP Version capable      : 1 to 3
VTP version running       : 1
VTP Domain Name           : cisco
VTP Pruning Mode          : Disabled
VTP Traps Generation      : Disabled
Device ID                 : 0c09.f972.8700
Configuration last modified by 0.0.0 at 5-4-20 03:33:08
Local updater ID is 0.0.0 (no valid interface found)

Feature VLAN:
-----
VTP Operating Mode        : Server
Maximum VLANs supported locally : 1005
Number of existing VLANs   : 6
Configuration Revision     : 3
MDS digest                : 0xFC 0x05 0xC0 0x82 0x
                            0xF4 0x35 0x5D 0x76 0x
SW4#
```





## Spanning Tree Protocol (STP)

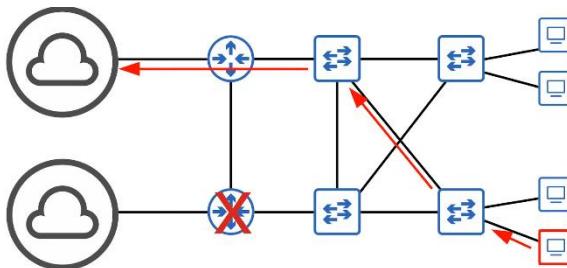
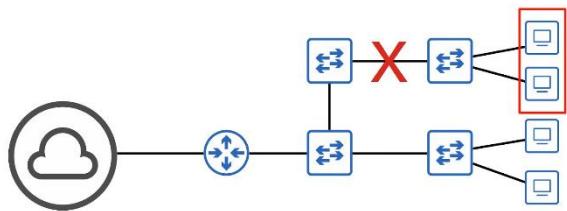
### STP (Part 1)

Topics covered:

- Redundancy in networks
- STP

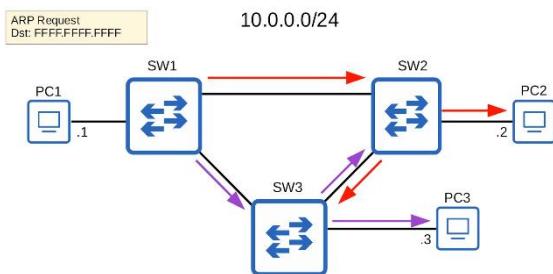
### Network Redundancy

- Essential part of network design
- Modern networks expected to run 24/7
- Need to design with redundancy in mind



- PCs usually only have 1 NIC, however servers usually have multiple

## Broadcast Storms



- When broadcast message sent out of PC1, the packet keeps on looping around and around
- In layer 3, the TTL header prevents this
- However, no TTL header in Ethernet field
- The broadcast frames will loop around the network indefinitely
- If too many, network will be too congested for legitimate traffic to use the network
- This is called a broadcast storm
- MAC Address Flapping
  - Each time a frame arrives on a switchport, the switch uses the source MAC address field to learn the MAC address and update its MAC address table

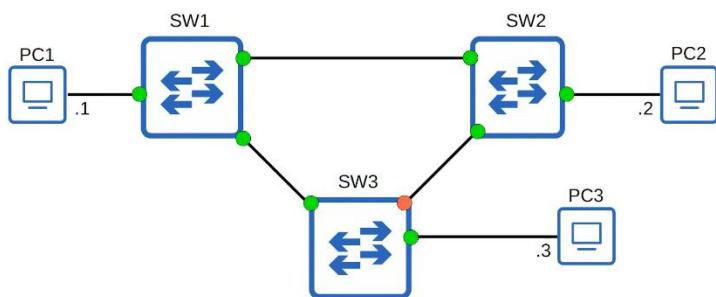
- When frames from the same source repeatedly arrive on different interfaces, the switch is continuously updating the interface in its MAC address table

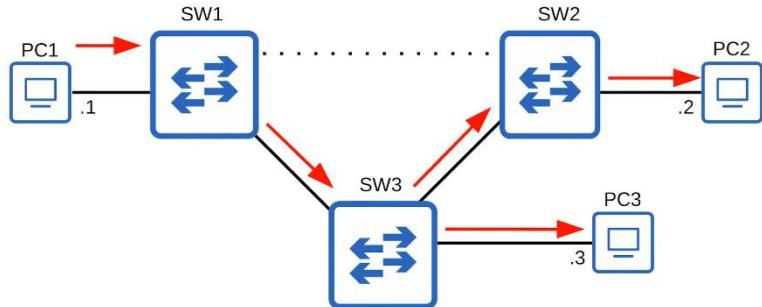
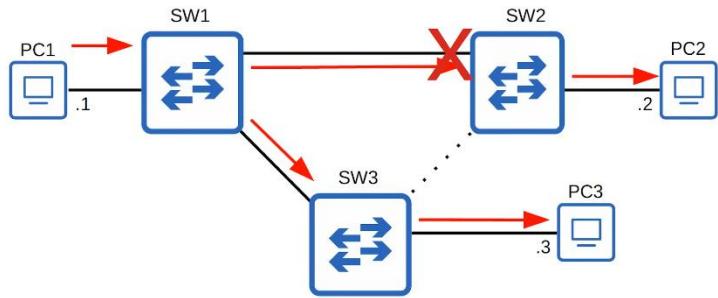
## Spanning Tree Protocol

- "Classic Spanning Tree Protocol" is IEEE 802.1D
- Switches from all vendors run STP by default
- Prevents Layer 2 loops by placing redundant ports in a blocking state, essentially disabling the interface
- These interfaces act as backups that can enter a forwarding state if an active (currently forwarding) interface fails
- Interfaces in a forwarding state behave normally. They send and receive traffic
- Interfaces in a blocking state only send or receive STP messages (called **BPDUs** = Bridge Protocol Data Units)

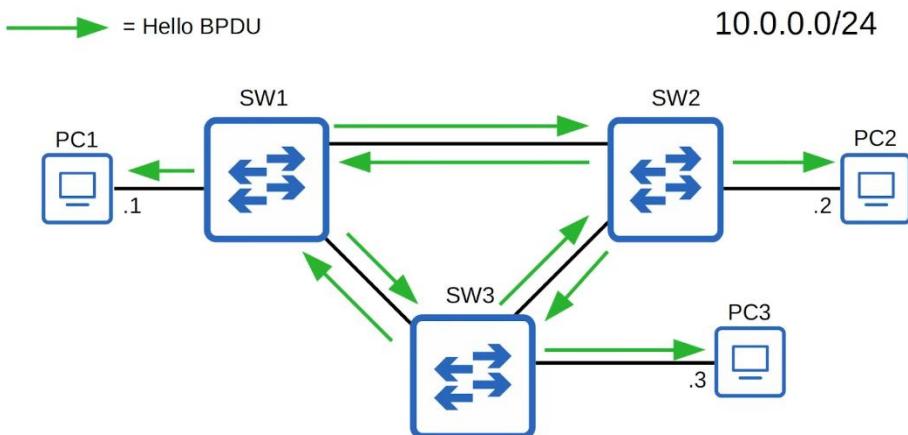


Spanning Tree Protocol still uses the term 'bridge'. However, when we use the term 'bridge', we really mean 'switch'. Bridges are not used in modern networks.





- By selecting which ports are forwarding and which are blocking, STP creates a single path to/from each point in the network
  - Prevents Layer 2 loops
- There is a set process that STP uses to determine which ports should be forwarding and which should be blocking
- STP-enabled switches send/receive Hello BPDUs out of all interfaces, the default timer is 2s, (switch will send Hello once every 2s)
- If a switch receives Hello BPDU on an interface, it knows that interface is connected to another switch (other devices don't use STP - don't send Hello BPDUs)

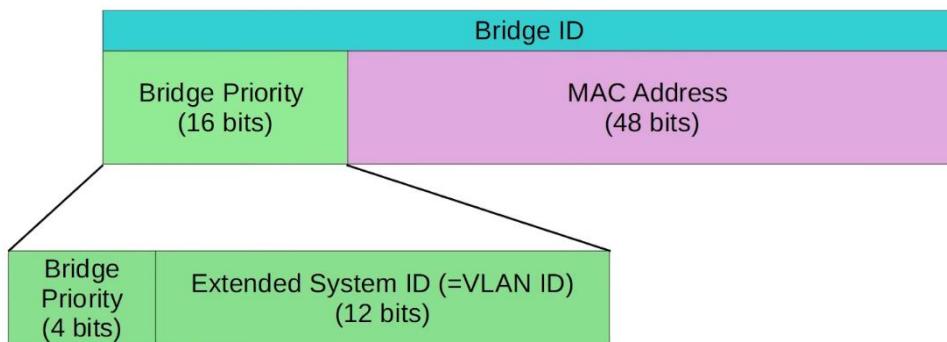


- Switches use 1 field in the STP BPDU, the Bridge ID field, to elect a root bridge for the network

- The switch with the lowest Bridge ID becomes the root bridge
- All ports on the root bridge are put in a forwarding state, and other switches in the topology must have a path to reach the root bridge



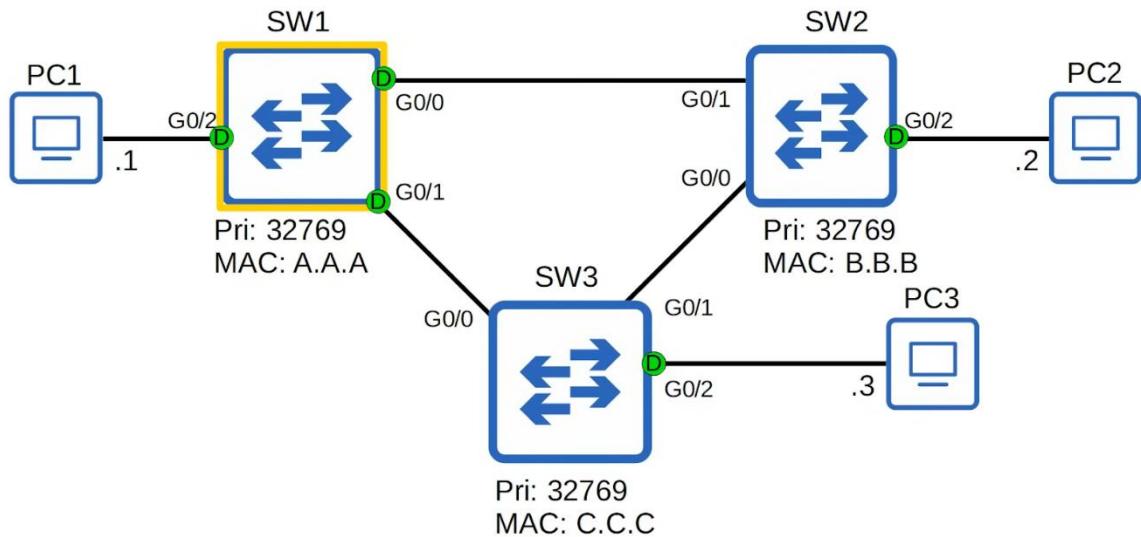
- Default bridge priority is 32768 on all switches
- So, the lowest MAC address is used as tie-breaker
- Bridge priority compared first before MAC address



- Cisco switches use a version of STP called PVST (Per-VLAN Spanning Tree)
- PVST runs a separate STP instance in each VLAN, so in each VLAN different interfaces can be forwarding/blocking

Bridge Priority				Extended System ID (VLAN ID)											
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

- Bridge priority + Extended System ID is a single field of Bridge ID
- Extended sys ID cannot be changed (determined by VLAN ID)
- Can only change bridge priority in units of 4096

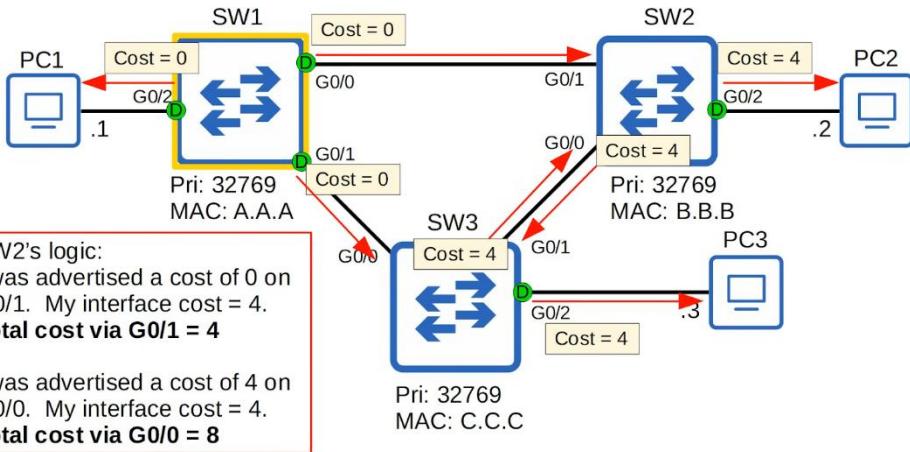


### Root bridges

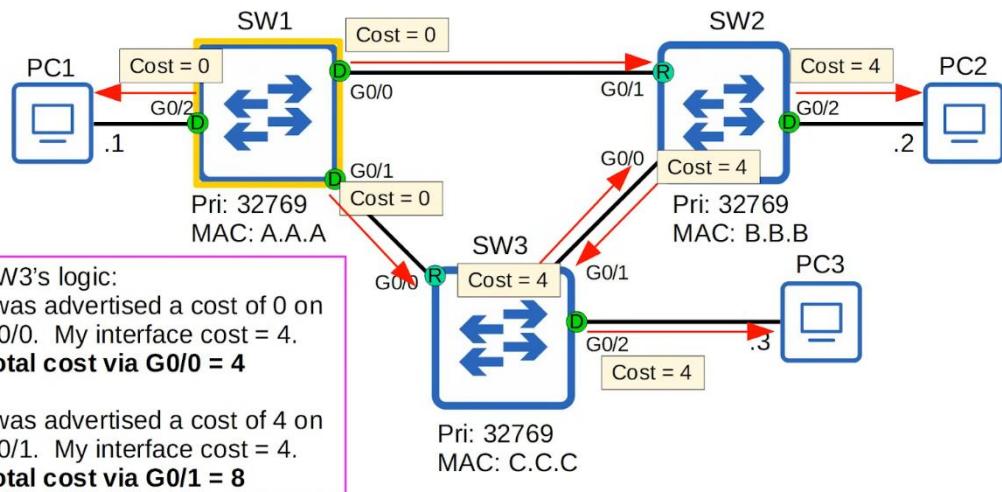
- When a switch is powered on, it assumes it is the root bridge
- It will give up position if it receives from a 'superior' BPDU (lower bridge ID)
- Once the topology has converged and all switches agree on the root bridge, only the root bridge sends BPDUs
- Other switches in the network will forward these BPDUs, but will not generate their own one

### Root Cost

Speed	STP Cost
10 Mbps	100
100 Mbps	19
1 Gbps	4
10 Gbps	2



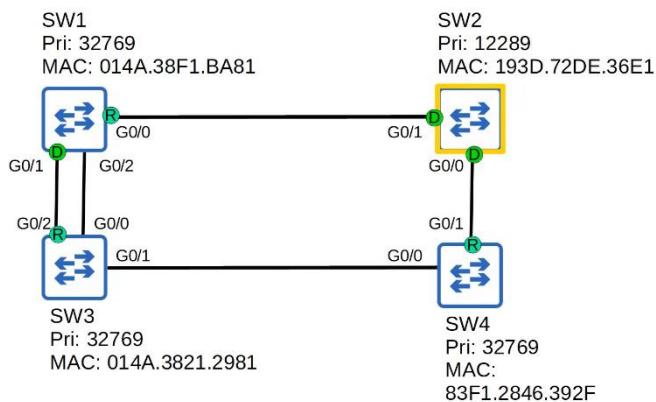
- SW2 will select G0/1 as root port



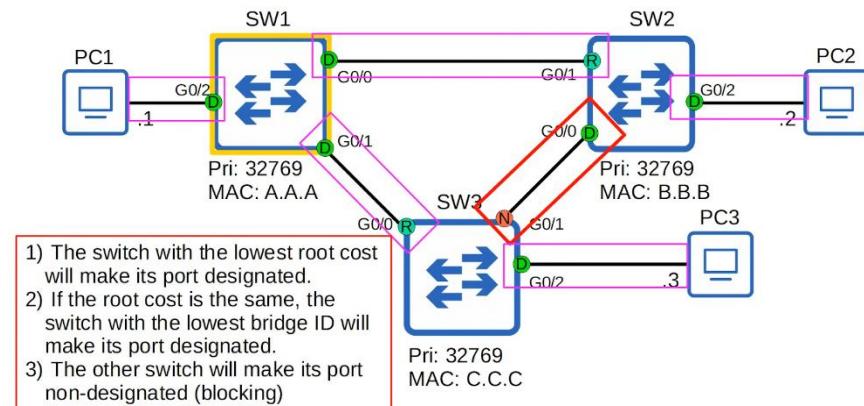
- Ports connected to another switch's root port MUST be designated.
  - Because the root port is the Switch's path to the root bridge, another switch must not block it

## Port ID

Sw1#show spanning-tree					
VLAN0001					
Spanning tree enabled protocol ieee					
Root ID	Priority	32769			
	Address	aaaa.aaaa.aaaa			
This bridge is the root					
Hello Time	2 sec	Max Age 20 sec	Forward Delay 15 sec		
Bridge ID	Priority	32769 (priority 32768 sys-id-ext 1)			
	Address	aaaa.aaaa.aaaa			
STP Port ID = port priority (default 128) + port number					
Interface	Role	Sts	Cost	Prio.Nbr	Type
G10/0	Dsg	FWD	4	128.1	Shr
G10/1	Dsg	FWD	4	128.2	Shr
G10/2	Dsg	FWD	4	128.3	Shr
G10/3	Dsg	FWD	4	128.4	Shr
G11/0	Dsg	FWD	4	128.5	Shr
G11/1	Dsg	FWD	4	128.6	Shr
G11/2	Dsg	FWD	4	128.7	Shr
G11/3	Dsg	FWD	4	128.8	Shr



Every collision domain has a single STP designated port.



## Steps for STP

1. The switch with the lowest bridge ID is elected as the root bridge. All ports on the root bridge are designated ports (forwarding state).

### Root bridge selection:

1. Lowest bridge ID

1. Each remaining switch will select 1 of its interface to be its root port. Interface with lowest root cost will be root port. Root ports are also in the forwarding state.

### **Root port selection:**

- 1. Lowest root cost**
  - 2. Lowest neighbour bridge**
  - 3. Lowest neighbour port ID**
- 
- 1. Each remaining collision domain will select 1 interface to be a designated port (forwarding state). The other port in the collision domain will be non-designated (blocking)**

### **Designated port selection:**

- 1. Interface on switch with the lowest root cost**
- 2. Interface on switch with lowest bridge ID**

When at step 3, ports connected to the root bridge will be non-designated

### **Commands**

- "show spanning tree"
- "show spanning tree vlan 1"
- "show spanning tree detail"
- "show spanning tree summary"

## **STP (Part 2)**

- Things covered
  - STP state/timers
  - STP BPDU
  - STP optional features
  - STP configuration

•

## STP States

STP Port State	Stable/Transitional
<b>Blocking</b>	Stable
<b>Listening</b>	Transitional
<b>Learning</b>	Transitional
<b>Forwarding</b>	Stable

- Root/Designated ports remains stable in a **Forwarding state**
- Non-designated ports remain stable in Blocking state
- Listening and Learning are transitional states which are passed through when an interface is activated, or when a Blocking port must transition to a Forwarding state due to a change in the network topology
- Blocking State
  - Non-designated ports are in a blocking state
  - Interfaces in a blocking state are effectively disabled to prevent loops
  - Interfaces do not send/receive regular network traffic
  - Interfaces receive BPDUs
  - Do not forward BPDUs
  - Do not learn MAC address
- Listening
  - After the blocking state, interfaces with the Designated or Root role enter the Listening state

- Only Designated or Root enter the Listening state (Non-designated ports are always Blocking)
- The Listening state is 15s by default. Determined by Forward delay timer
- An interface in the Listening state ONLY forwards/receives BPDUs
- An interface in the Listening state does NOT send/receive regular traffic
- Does NOT learn MAC address from regular traffic that arrives on the interface

- Learning

- After Listening state, a Designated or Root port will enter the Learning state
- 15s by default, determined by Forward delay timer (same used in Listening state)
- An interface ONLY send/receive BPDUs
- Does NOT send/receive regular traffic
- Learns MAC address from regular traffic that arrives on the interface

- Forwarding

- Roots and Designated ports are in Forwarding state
- Port in the Forwarding state operate as normal
- Send/receives BPDUs
- Send/receives normal traffic
- Learns MAC addresses

STP Port State	Send/Receive BPDUs	Frame forwarding (regular traffic)	MAC address learning	Stable/Transitional
<b>Blocking</b>	NO/YES	NO	NO	Stable
<b>Listening</b>	YES/YES	NO	NO	Transitional
<b>Learning</b>	YES/YES	NO	YES	Transitional
<b>Forwarding</b>	YES/YES	YES	YES	Stable
<b>Disabled</b>	NO/NO	NO	NO	Stable

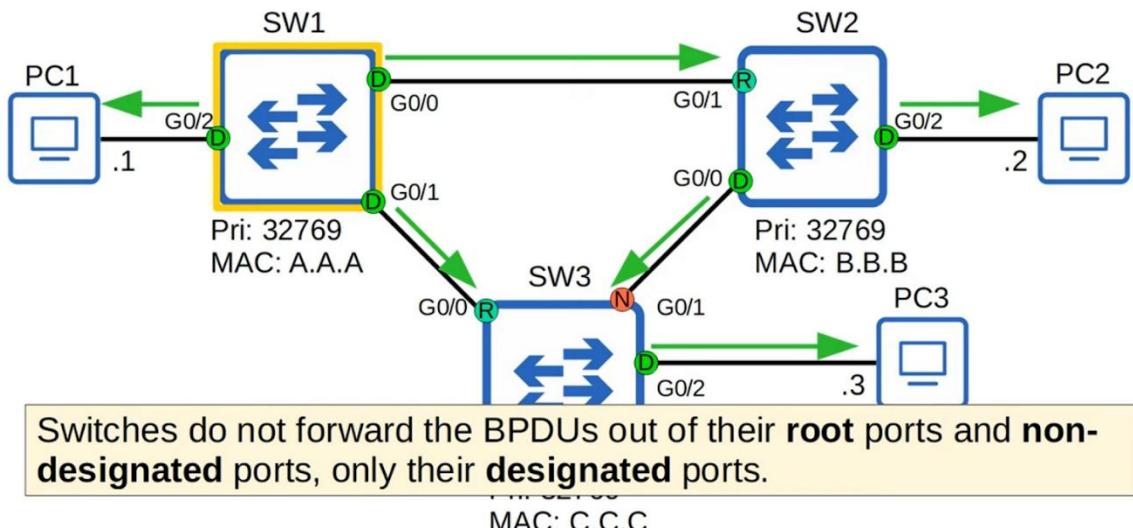
- 

## Spanning Tree Timers

STP Timer	Purpose	Duration
<b>Hello</b>	How often the root bridge sends hello BPDUs	2sec
<b>Forward delay</b>	How long the switch will stay in the Listening and Learning states (each state is 15 seconds = total 30 seconds)	15sec
<b>Max Age</b>	How long an interface will wait <u>after ceasing to receive Hello BPDUs</u> to change the STP topology.	20sec (10* hello)

- Max Age
  - If BPDU is received before max age, timer reset to 20s
  - If not received, switch will reevaluate its STP choices, including root bridge, local root, and non-designated ports
  - If a non-designated port is selected to become a designated/root port, it will transition from the blocking state to the listening state (15s), learning state (15s), and then the forwarding state. Can take 50s for a blocking interface to transition to forwarding
  - These timers and transitional states are to make sure loops aren't created due to interface moving to forwarding state too soon
  - But a forwarding state can go straight to blocking state
- The STP timers on the root bridge determine the timer for the whole network

→ = Hello BPDU



- Ethernet Destination MAC address
  - PVST - only ISL trunk encapsulation
    - 01:00:0c:cc:cc:cd
  - PVST+ - supports 802.1q
    - 0180:c2000:0000

### STP Optional Features (STP Toolkit)

- Portfast
  - Allows port to move immediately to the Forwarding state, bypassing Listening and Learning (not waiting 30s)
  - Must be enabled only on ports connected to end hosts
  - Else, if connected to switch, can cause layer 2 loops

```

SW1(config)#interface g0/2
SW1(config-if)#spanning-tree portfast
%Warning: portfast should only be enabled on ports connected to a single
host. Connecting hubs, concentrators, switches, bridges, etc... to this
interface when portfast is enabled, can cause temporary bridging loops.
Use with CAUTION

%Portfast has been configured on GigabitEthernet0/2 but will only
have effect when the interface is in a non-trunking mode.
SW1(config-if)#

```

- SW1(config)# **spanning-tree portfast default**
  - Enable portfast on all access ports, not trunk ports

- BPDU Guard
  - If an interface receives a BPDU from another switch, the interface will shutdown to prevent a loop from forming
    - Occurs when instead of connecting to end host, connect to switch

```
SW1(config)#interface g0/2
SW1(config-if)#spanning-tree bpduguard enable
SW1(config-if)#[REDACTED]
```

- SW1(config)# **spanning-tree portfast bpduguard default**
  - Enables BPDU guard on all portfast enabled interface

<b>Root Guard</b>	If you enable <b>root guard</b> on an interface, even if it receives a superior BPDU (lower bridge ID) on that interface, the switch will not accept the new switch as the root bridge. The interface will be disabled.
<b>Loop Guard</b>	If you enable <b>loop guard</b> on an interface, even if the interface stops receiving BPDUs, it will not start forwarding. The interface will be disabled.

- Probably wont need to know for CCNA

## Spanning Tree Configuration

```
SW1(config)#spanning-tree mode ?
  mst      Multiple spanning tree mode
  pvst     Per-Vlan spanning tree mode
  rapid-pvst Per-Vlan rapid spanning tree mode

SW1(config)#spanning-tree mode pvst
SW1(config)#[REDACTED]
```

```

SW3(config)#spanning-tree vlan 1 root primary
SW3(config)#do show spanning-tree

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    24577
              Address     cccc.cccc.cccc
              This bridge is the root
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    24577  (priority 24576 sys-id-ext 1)
              Address     cccc.cccc.cccc
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time   15  sec

```

The **spanning-tree vlan *vlan-number* root primary** command sets the STP priority to 24576. If another switch already has a priority lower than 24576, it sets this switch's priority to 4096 less than the other switch's priority.

- STP priority is bridge priority

```

SW2(config)#spanning-tree vlan 1 root secondary
SW2(config)#do show spanning-tree

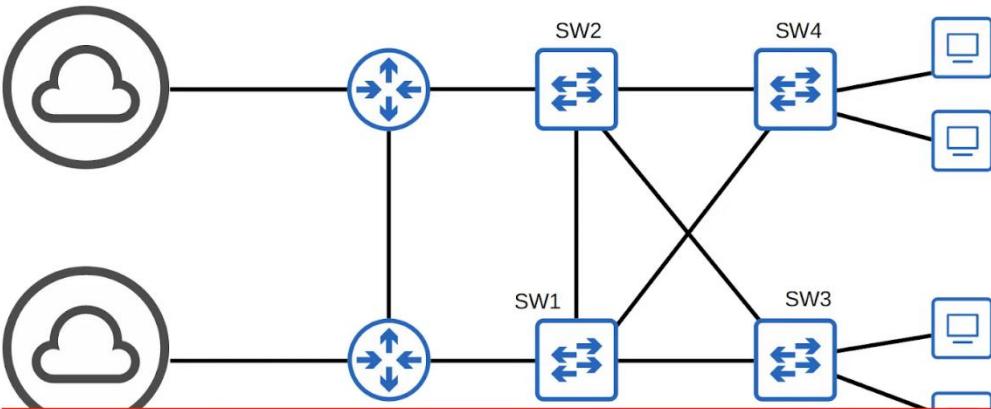
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    24577
              Address     cccc.cccc.cccc
              Cost         4
              Port        1 (GigabitEthernet0/0)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    28673  (priority 28672 sys-id-ext 1)
              Address     bbbb.bbbb.bbbb
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time   300 sec

```

The **spanning-tree vlan *vlan-number* root secondary** command sets the STP priority to 28672.

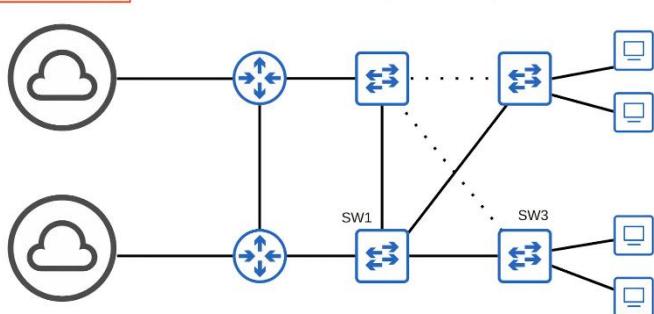
## Spanning Tree Quiz 7



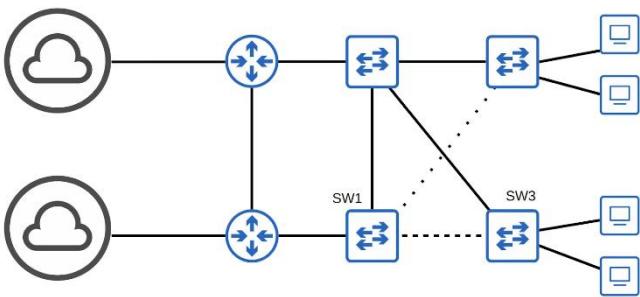
Two VLANs are active in this network, 10 and 20. By default, SW3 is the root bridge for both VLANs. Configure SW1 as the primary root for VLAN10 and the secondary root for VLAN20. Configure SW2 as the primary root for VLAN20 and the secondary root for VLAN10. Which two commands should you issue on SW1, and which two commands should you issue on SW2?

- SW1
  - "spanning-tree vlan 10 root primary"
  - "spanning-tree vlan 20 root secondary"
- SW2
  - "spanning-tree vlan 10 secondary"
  - "spanning-tree vlan 20 primary"

VLAN 10 Topology



### VLAN 20 Topology



### Port settings

```
SW2(config-if)#spanning-tree vlan 1 ?
  cost          Change an interface's per VLAN spanning tree path cost
  port-priority Change an interface's spanning tree port priority

SW2(config-if)#spanning-tree vlan 1 cost ?

<1-200000000>  Change an interface's per VLAN spanning tree path cost

SW2(config-if)#spanning-tree vlan 1 cost 200
SW2(config-if)#spanning-tree vlan 1 port-priority ?
  <0-224>  port priority in increments of 32

SW2(config-if)#spanning-tree vlan 1 port-priority 32
SW2(config-if)#[
```

## STP Additional Features

### PortFast



```
SW1(config)# interface g0/1
SW1(config-if)# spanning-tree portfast
%Warning: portfast should only be enabled on ports connected to a single
host. Connecting hubs, concentrators, switches, bridges, etc... to this
interface when portfast is enabled, can cause temporary bridging loops.
Use with CAUTION

%Portfast has been configured on GigabitEthernet0/1 but will only
have effect when the interface is in a non-trunking mode.

SW1# show spanning-tree interface g0/1 detail
Port 2 (GigabitEthernet0/1) of VLAN001 is designated forwarding
  Port path cost 4, Port priority 128, Port Identifier 128.2.
  Designated root has priority 32769, address 5254.0016.c410
  Designated bridge has priority 32769, address 5254.0016.c410
  Designated port id is 128.3, designated path cost 0
  Timers: message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state: 1
  The port is in the portfast edge mode
  Link type is point-to-point by default
  BPDU: sent 1272, received 0
```

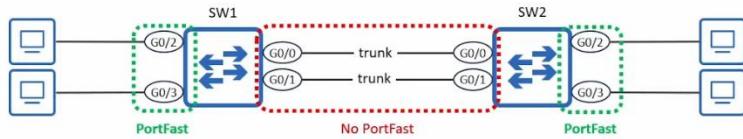
PortFast should NOT be configured on ports connected to switches or temporary Layer 2 loops can occur.

Even if you configure **spanning-tree portfast** on a trunk port, it won't be active.

show spanning-tree interface interface-name detail

- There are two kinds of PortFast:
  - edge**
  - network**
- edge** is the kind we are covering in this video.
- network** is used for a feature called Bridge Assurance (not a CCNA topic).

- 



```
SW1(config)# spanning-tree portfast default
%Warning: this command enables portfast by default on all interfaces. You
should now disable portfast explicitly on switched ports leading to hubs,
switches and bridges as they may create temporary bridging loops.
```

- Access ports only (not trunk ports).
- To disable PortFast on a specific access port:
  - SW1(config-if)# spanning-tree portfast disable

```
SW1# show spanning-tree interface g0/2 detail
Port 3 (GigabitEthernet0/2) of VLAN0001 is designated forwarding
  Port path cost 4, Port priority 128, Port Identifier 128.3.
  Designated root has priority 32769, address 5254.0016.c410
  Designated bridge has priority 32769, address 5254.0016.c410
  Designated port id is 128.3, designated path cost 0
  Timers: message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state: 4
  The port is in the portfast edge mode by default
  Link type is point-to-point by default
  BPDU: sent 22, received 0
```

```
SW1# show spanning-tree interface g0/1 detail
Port 2 (GigabitEthernet0/1) of VLAN0001 is designated blocking
  Port path cost 4, Port priority 128, Port Identifier 128.2.
  Designated root has priority 32769, address 5254.0016.c410
  Designated bridge has priority 32769, address 5254.0016.c410
  Designated port id is 128.2, designated path cost 0
  Timers: message age 0, forward delay 10, hold 0
  Number of transitions to forwarding state: 0
  Link type is point-to-point by default
  BPDU: sent 17, received 0
```

- In some cases, you might want to enable PortFast on a trunk
  - Port connected to a virtualization server with VMs in different VLANs
  - Port connected to router via router-on-a-stick (ROAS)
- Can only be configured per port
  - SW1(config-if)# spanning tree portfast trunk

## PortFast Edge



## PortFast edge



- In modern Cisco switches, if you use the commands covered in this lecture, the device will automatically add the **edge** keyword to the configuration.

- SW1(config-if)# **spanning-tree portfast**
- In the running-config: **spanning-tree portfast edge**
- SW1(config-if)# **spanning-tree portfast trunk**
- In the running-config: **spanning-tree portfast edge trunk**
- SW1(config)# **spanning-tree portfast default**
- In the running-config: **spanning-tree portfast edge default**

- You can use either version of the commands when configuring PortFast.
- The end result is the same: **edge** will always be added in the configuration.
- **spanning-tree portfast disable** doesn't use the **edge** keyword.

```
SW1(config)# interface g0/1
SW1(config-if)# spanning-tree portfast
SW1# show running-config interface g0/1
Building configuration...
Current configuration : 113 bytes
!
interface GigabitEthernet0/1
  switchport access vlan 10
  switchport mode access
  spanning-tree portfast edge
end
```

→ Use **show running-config interface interface-name** to view the running-config for the specified interface.  
→ Doesn't work in Packet Tracer.

→ Automatically added **edge** to the end of the command.

## Summary

- When a host connects to a switch port, by default, it takes 30s before port can send/receive data
- PortFast allows a switch port to immediately enter the STP Forwarding state, bypassing the Listening and Learning states
- It can be configured 2 ways
  - SW1(config-if)# **spanning-tree portfast [edge]**
  - SW1(config)# **spanning-tree portfast [edge] default**
- Can use "spanning-tree portfast disable" to disabled specific ports
- Portfast should not be configured on ports connected to a switch, as it can cause temporary layer 2 loops
- Can enable on trunks using, "**spanning-tree portfast [edge] trunk**"

## BPDUs Guard

- If a portfast enabled port receives a BPDU (it should not receive since portfast enabled should only be connected to non-switch), it will revert to acting like a normal STP port (no portfast)

- If BPDU Guard enabled port receives a BPDU, it enters error-disabled state (disables the port)
- Config
  - SW1(config-if)# **spanning-tree bpduguard enable**
  - SW1(config)# **spanning-tree portfast [edge] bpduguard default**
    - Enabled only on all portfast enabled ports
  - SW1(config-if)# **spanning-tree bpduguard disable**
- Enabling the error-disabled port
  - Fix the underlying issue
  - Manual
    - Shutdown and no shutdown
  - Auto
    - ErrDisable Recovery
- ErrDisable Recovery
  - Feature that auto re-enable err-disabled ports after a certain period of time

```
SW3# show errdisable recovery
ErrDisable Reason           Timer Status
-----
arp-inspection
bpduguard                  Disabled
channel-misconfig (STP)
!output omitted
Timer interval: 300 seconds
Interfaces that will be enabled at the next timeout:
```

- Disabled by default
- SW3(config)# **errdisable recovery cause <cause>**
  - <cause>: bpduguard
  - Enable bpduguard
- SW3(config)# **errdisable recovery interval <interval>**
  - Set the timing interval in seconds

- Default time is 300s / 5mins

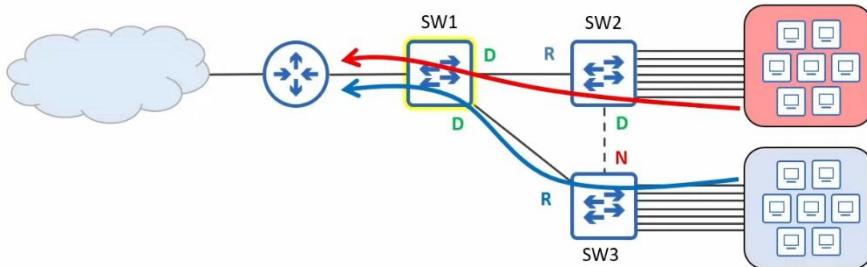
## BPDU Filter

- Problem
  - Switch connected to not-switch (e.g. PC) sends BPDU
  - Waste processing power and bandwidth
  - Security concern
- BPDU Filter stops a port from sending BPDUs
  - However, doesn't disable the port if receives a BPDU
- Config
  - SW3(config-if)# **spanning-tree bpdu filter**
    - Port will not send and ignore BPDUs received
    - Disables STP on the port, USE WITH CAUTION
  - SW3(config)# **spanning-tree portfast [edge] bpdufilter default**
    - BPDU filter enabled on ports with portfast
    - "**spanning-tree bpdufilter disable**" to disable on specific ports
    - Will not send BPDUs
    - If receives BPDUs, PortFast and BPDU Filter disabled, acts as normal STP port
- BPDU Guard and Filter can be applied at the same time
  - When BPDU is received
  - If filter applied on global config mode
    - BPDU guard will be triggered (err-disable port)
  - If filter applied on interface mode
    - BPDU ignored
    - BPDU guard will not be triggered
- Recommendation:
  - BPDU guards can use any method

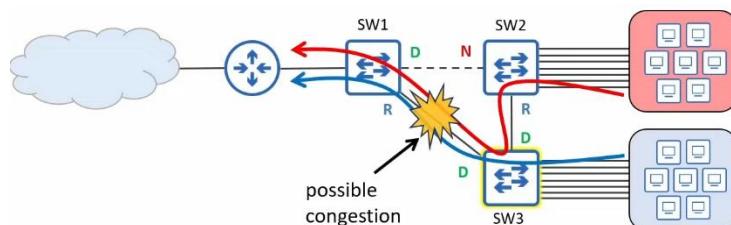
- BPDU Filter, use global mode

## Root Guard

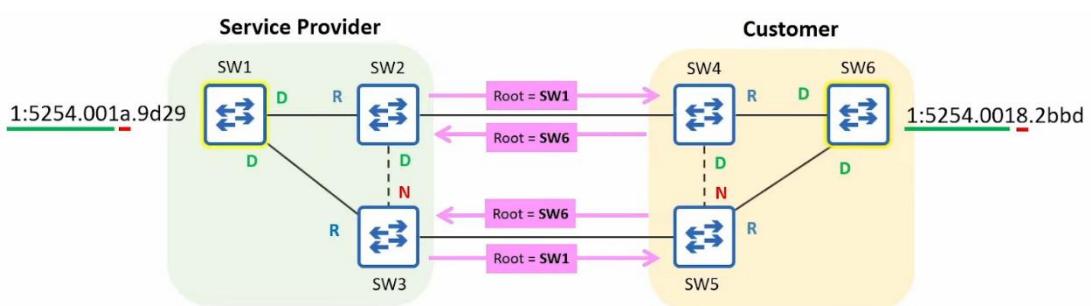
- Root bridge placement



- When electing root bridge, should consider
  - Optimal traffic flow
  - Stability & reliability (use the best switch for root bridge)

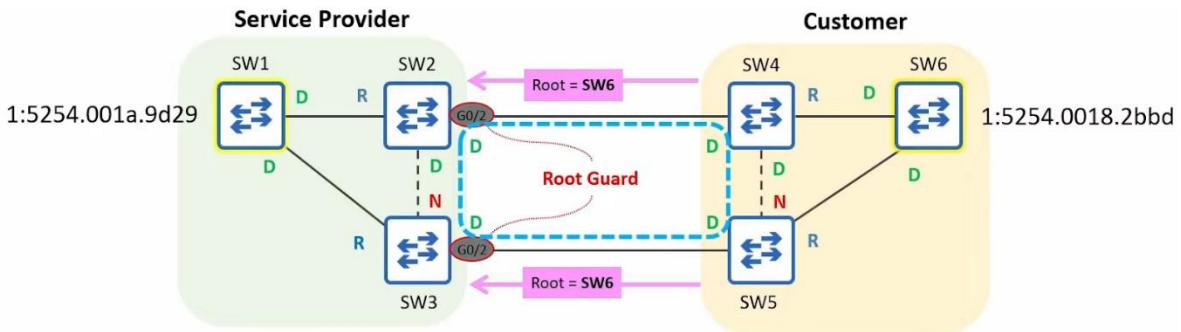


- Problem

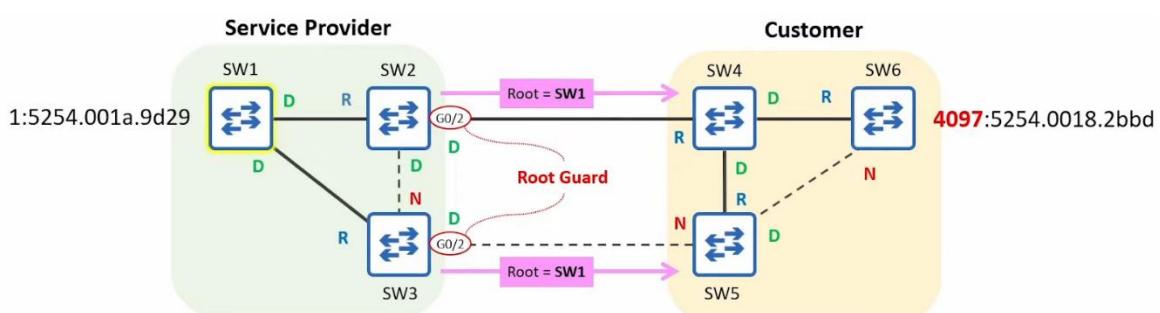


- Customer root switch becomes the root switch because even though same priority, MAC address lower
- Root Guard prevents this
- Root Guard config
  - Only in interface mode

- Interface the ports connected to the other network
- SW2(config-if)# **spanning-tree guard root**
- If root-guard enabled port receives BPDUs, will enter Broken(Root Inconsistent) state, effectively disabling it



- To re-enable
  - Disabled port must stop receiving superior BPDUs
  - Tell customer to change the priority of their switch
- Once the superior BPDUs received by SW2 G0/2 and SW3 G0/3 age out, the ports will auto re-enable
  - BPDU max age is 20s by default

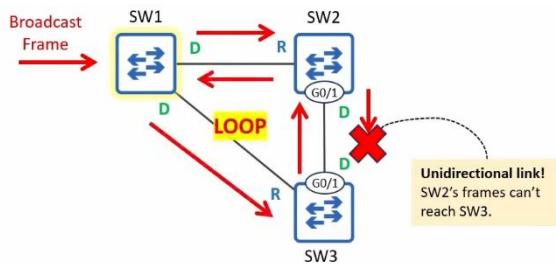


- Note
  - Customer should NOT enable root guard

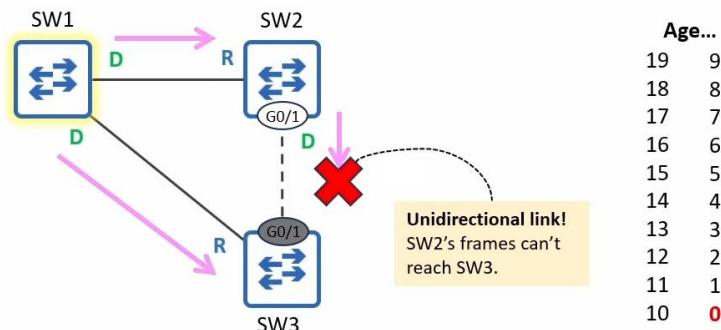
## Loop Guard

- Uni directional link
  - Normally caused by layer 1 issue
    - Damaged cable

- Faulty connectors/transceivers (e.g. SFP)
- Common in fibre optics compared to UTP
  - Fibre optic use 2 separate fibres
  - If 1 damaged, can cause unidirectional link
- Problem



- SW3 G0/1 never receive BPDU
  - Will become Designated port
  - Start forwarding BPDUs
- Loop formed
- Solution
  - When Loop-Guard enabled port's max age timer reaches 0, it doesn't become a Designated port and start transitioning to Forwarding
    - Enters Broken (Loop Inconsistent) state
    - Port remain up/up, but STP blocks it
  - To re-enable
    - When the broken port start receiving BPDUs, auto re-enable



- Config
  - SW3(config-if)# **spanning-tree guard loop**

- SW3(config)# **spanning-tree loopguard default**
  - Enable on all ports
  - SW3(config-if)# spanning-tree loopguard none (disable for specific interface)
- Note
  - Loop guard should be enabled on root and non-designated ports (ports that are supposed to receive BPDUs)
  - Loop guard and root guard are mutually exclusive
    - Cannot enable both on the same port at the same time
    - Global vs Interface cmd -> Interface cmd takes effect

## Rapid Spanning Tree

Things covered

- Comparison of STP versions
- Rapid PVST+

Spanning Tree Versions	
Industry standards (IEEE)	Cisco versions
<b>Spanning Tree Protocol (802.1D)</b> <ul style="list-style-type: none"> <li>• The original STP</li> <li>• All VLANs share one STP instance.</li> <li>• Therefore, cannot load balance.</li> </ul>	<b>Per-VLAN Spanning Tree Plus (PVST+)</b> <ul style="list-style-type: none"> <li>• Cisco's upgrade to 802.1D</li> <li>• Each VLAN has its own STP instance.</li> <li>• Can load balance by blocking different ports in each VLAN.</li> </ul>
<b>Rapid Spanning Tree Protocol (802.1w)</b> <ul style="list-style-type: none"> <li>• Much faster at converging/adapating to network changes than 802.1D</li> <li>• All VLANs share one STP instance.</li> <li>• Therefore, cannot load balance.</li> </ul>	<b>Rapid Per-VLAN Spanning Tree Plus (Rapid PVST+)</b> <ul style="list-style-type: none"> <li>• Cisco's upgrade to 802.1w</li> <li>• Each VLAN has its own STP instance.</li> <li>• Can load balance by blocking different ports in each VLAN.</li> </ul>
<b>Multiple Spanning Tree Protocol (802.1s)</b> <ul style="list-style-type: none"> <li>• Uses modified RSTP mechanics.</li> <li>• Can group multiple VLANs into different instances (ie. VLANs 1-5 in instance 1, VLANs 6-10 in instance 2) to perform load balancing.</li> </ul>	

## Rapid Spanning Tree Protocol

- Similarities btw STP and RSTP
  - RSTP serves the same purpose as STP, blocking specific ports to prevent layer 2 loops
  - RSTP elects a root bridge with the same rules as STP
  - RSTP elects a root port with the same rules as STP
  - RSTP elects a designated port with the same rules as STP
- RSTP cost

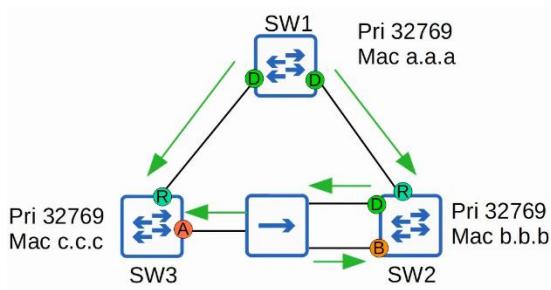
Speed	STP Cost	RSTP Cost
10 Mbps	100	2,000,000
100 Mbps	19	200,000
1 Gbps	4	20,000
10 Gbps	2	2000
100 Gbps	X	200
1 Tbps	X	20

- RSTP States

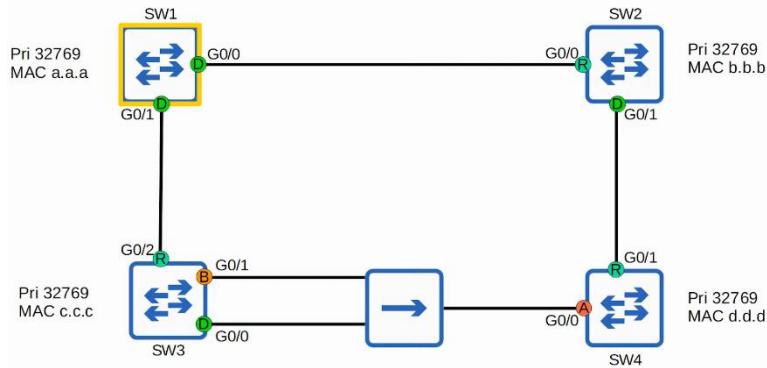
STP Port State	Send/Receive BPDUs	Frame forwarding (regular traffic)	MAC address learning	Stable/Transitional
Discarding	NO/YES	NO	NO	Stable
Learning	YES/YES	NO	YES	Transitional
Forwarding	YES/YES	YES	YES	Stable

- If a port is administratively down (shutdown command) = discarding state
  - Previously disabled state
- If a port is enabled but blocking traffic to prevent layer 2 loops = discarding state

- Previously blocking state
  
- RSTP Port Roles
  - Root port role remains unchanged
  - Designated port role remains unchanged
  - Non-designated port role is split into 2:
    - Alternate port
    - Backup port
  - Alternate port
    - It is a discarding port that receives a superior BPDU from another switch
    - Same as blocking ports in STP
    - Functions as backup to root port
    - If root port fails, the switch can immediately move its best alternate port forward
    - Functions like optional STP feature - UplinkFast, but built into RSTP already
      - RSTP: Backbone functionality (Not in CCNA)
        - Optional STP feature built into RSTP
        - Allows the switch to expire the max age timer on its interface and rapidly forward the superior BPDUs
        - Built in so don't need configure
  - Backup port
    - It is a discarding port that receives a superior BDPU from another interface on the same switch
    - This collision only occurs when 2 interfaces are connected to the same collision domain (via a hub)
    - Hubs are not used nowadays so will not encounter a backup port



### Practice Q



### Inter-operability

- Rapid STP is compatible with Classic STP
- Interfaces on RSTP-enabled switch connected to STP-enabled switch will operate in STP mode (timers, blocking-> listening->learning->forwarding process etc)

### RSTP BPDU

```

> Frame 999: 68 bytes on wire (544 bits), 68 bytes captured (544 bits)
> Ethernet II, Src: aa:aa:aa:aa:ab (aa:aa:aa:aa:ab), Dst: PVST+ (00:0c:29:00:00:00)
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 10
> Logical-Link Control
< Spanning Tree Protocol
  Protocol Identifier: Spanning Tree Protocol (0x0000)
  Protocol Version Identifier: Spanning Tree (0)
  BPDU Type: Configuration (0x00)
< BPDU Flags: 0x00
  0... .... = Topology Change Acknowledgment: No
  .... ..0 = Topology Change: No
< Root Identifier: 32768 / 10 / aa:aa:aa:aa:aa
  Root Bridge Priority: 32768
  Root Bridge System ID Extension: 10
  Root Bridge System ID: aa:aa:aa:aa:aa (aa:aa:aa:aa:aa)
  Root Path Cost: 0
  Bridge Identifier: 32768 / 10 / aa:aa:aa:aa:aa
  Bridge Priority: 32768
  Bridge System ID Extension: 10
  Bridge System ID: aa:aa:aa:aa:aa (aa:aa:aa:aa:aa)
  Port Identifier: 0x8002
  Message Age: 0
  Max Age: 20
  Hello Time: 2
  Forward Delay: 15
> Frame 71: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
> IEEE 802.3 Ethernet
> Logical-Link Control
< Spanning Tree Protocol
  Protocol Identifier: Spanning Tree Protocol (0x0000)
  Protocol Version Identifier: Rapid Spanning Tree (2)
  BPDU Type: Rapid/Multiple Spanning Tree (0x02)
< BPDU Flags: 0x3c, Forwarding, Learning, Port Role: Designated
  0... .... = Topology Change Acknowledgment: No
  .0... .... = Agreement: No
  ..1 .... = Forwarding: Yes
  ...1 .... = Learning: Yes
  .... 11.. = Port Role: Designated (3)
  .... ..0. = Proposal: No
  .... ..0 = Topology Change: No
< Root Identifier: 32768 / 1 / aa:aa:aa:aa:aa
  Root Bridge Priority: 32768
  Root Bridge System ID Extension: 1
  Root Bridge System ID: aa:aa:aa:aa:aa (aa:aa:aa:aa:aa)
  Root Path Cost: 4
< Bridge Identifier: 32768 / 1 / cc:cc:cc:cc:cc:cc
  Bridge Priority: 32768
  Bridge System ID Extension: 1
  Bridge System ID: SiliconL_cc:cc:cc (cc:cc:cc:cc:cc:cc)
  Port identifier: 0x8001
  Message Age: 1
  Max Age: 20
  Hello Time: 2
  Forward Delay: 15
  Version 1 Length: 0

```

- Differences

- Protocol Version
  - STP - 0
  - RSTP - 2
- BPDU Type
  - STP - 0
  - RSTP - 2
- BPDU Flags
  - STP uses 2 bits
  - RSTP uses all 8 bits
- In STP, only the root bridge originate the BPDUs, and other switches just forward the BPDUs received
- In RSTP, ALL switches originate and send their own BPDUs from designated ports

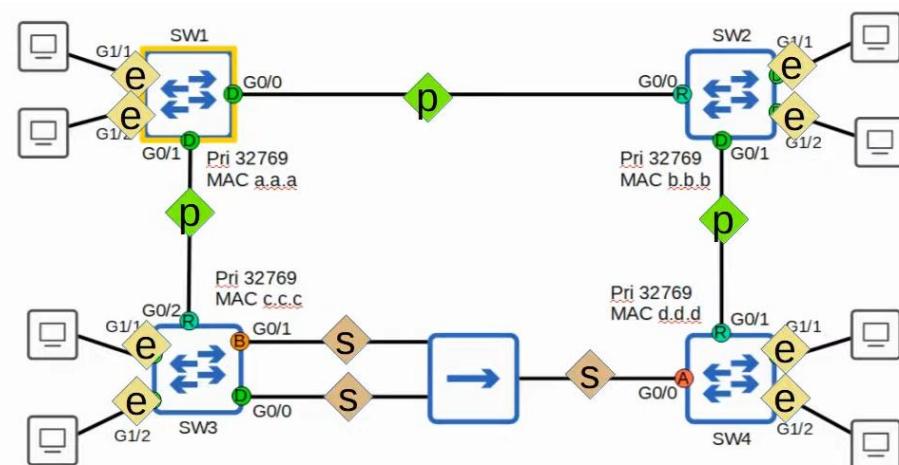
## Differences

- All switch running RSTP send their own BPDUs every hello time (2s)
- Switches 'age' the BPDU information much more quickly
  - In STP, a switch waits 10 hello intervals (20s)

- In RSTP, a switch considers a neighbour lost if it misses 3 BPDUs (6s)
- It will then 'flush' all MAC addresses learned on the interface as unable to reach them from that interface

## RSTP Link Types

- RSTP distinguishes btw 3 different 'link types'
  - Edge
    - Port that is connected to end host
    - Moves directly to forwarding, without negotiation
    - Need to enable "SW1(config-if)# **spanning-tree portfast**"
  - Point-to-point
    - Direct connection btw 2 switches
    - Full-duplex
    - Don't need to configure, will auto detect
    - SW1(config-if)# **spanning-tree link-type point-to-point**
  - Shared
    - Shared ports connected to another switch(es) via a hub
    - Must operate in half-duplex
    - Don't need config, will auto detect
    - SW1(config-if)# **spanning-tree link-type shared**



## Summary

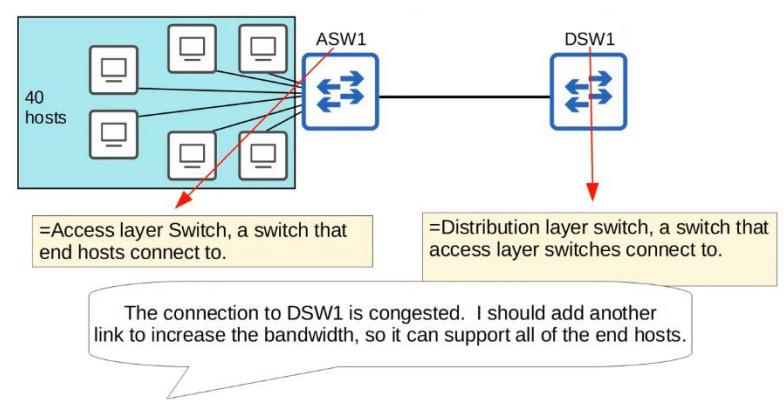
- Comparison of STP versions (standard vs Cisco)
- Rapid PVST+
  - RSTP port states (discarding, learning, forwarding)
  - RSTP port roles (root, designated, alternate, backup)
  - STP optional features built into RSTP (UplinkFast, BackboneFast, PortFast)
  - RSTP BPDU (sent by all switches, not just the root bridge)
  - RSTP link types (edge, point-to-point, shared)

## EtherChannel

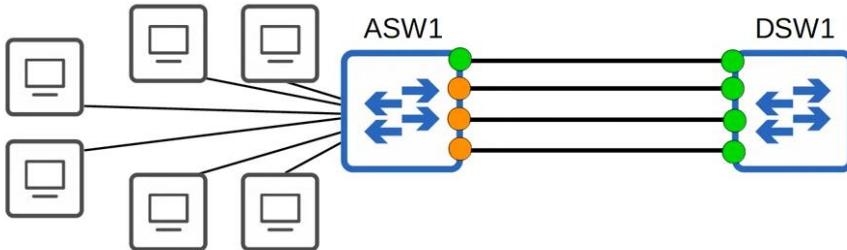
### Things Covered

- What is EtherChannel? What problems does it solve?
- Configuring Layer 2/3 EtherChannels

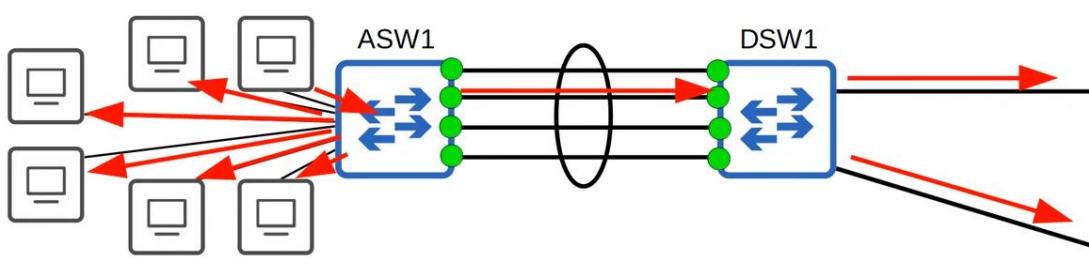
### EtherChannel



- Oversubscription
  - When bandwidth of interfaces connected to end hosts > bandwidth of connection of distribution switches
  - Some oversubscription is acceptable, but too much will cause congestion



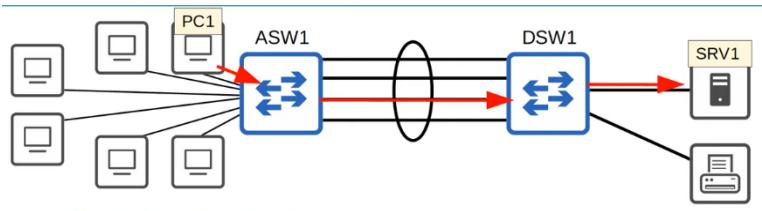
- If you connect 2 switches together with multiple links, all except one will be disabled by spanning tree
- If all of ASW1's interfaces were forwarding, layer 2 loops will form
- Other links will not be used unless active link fails. In that case, one of the inactive links will start forwarding



- EtherChannel groups multiple interfaces together to act as a single interface
- Represented by circle
- STP will treat this group as a single interface
- When a PC sends a broadcast message, ASW1 will broadcast to all other PCs and only 1 interface connected to DSW1
- Traffic using the EtherChannel will be load balanced among the physical interfaces in the group
- Common names
  - Port Channel

- LAG (Link Aggregation Group)

## Load Balancing



- EtherChannel load balances based on 'flows'
- A flow is a communication btw 2 nodes in the network
- Frames in the same flow will be forwarded using the same physical interface
- If frames in the same flow were forwarded using different physical interfaces, some frames may arrive at the destination out of order, which can cause problems
- E.g. PC1 -> SRV1, will always use the same interface
- You can change the inputs used in the interface selection calculation
- Inputs that can be used (e.g. src MAC, the same source MAC will use the same route)
  - Src MAC
  - Dst MAC
  - Src & Dst MAC
  - Src IP
  - Dst IP
  - Src & Dst IP
- The types of inputs that is supported by the switch depends on the type of switch
- Configuration of load-balance

```

ASW1#show etherchannel load-balance
EtherChannel Load-Balancing Configuration:
    src-dst-ip

EtherChannel Load-Balancing Addresses Used Per-Protocol:
Non-IP: Source XOR Destination MAC address
IPv4: Source XOR Destination IP address
IPv6: Source XOR Destination IP address

ASW1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
ASW1(config)#port-channel load-balance src-dst-mac
ASW1(config)#do show etherchannel load-balance
EtherChannel Load-Balancing Configuration:
    src-dst-mac

EtherChannel Load-Balancing Addresses Used Per-Protocol:
Non-IP: Source XOR Destination MAC address
IPv4: Source XOR Destination MAC address
IPv6: Source XOR Destination MAC address

ASW1(config)#

```

<pre> ASW1(config)#port-channel load-balance ? dst-ip      Dst IP Addr dst-mac     Dst Mac Addr src-dst-ip Src XOR Dst IP Addr src-dst-mac Src XOR Dst Mac Addr src-ip      Src IP Addr src-mac     Src Mac Addr </pre>	<pre> ASW1(config)#port-channel load-balance </pre>
---	---

- ASW1# **show etherchannel load-balance**
- ASW1(config)# **port-channel load-balance <method>**
  - <method>
    - **dst-ip**
    - **dst-mac**
    - **src-dst-ip**
    - **src-dst-mac**
    - **src-ip**
    - **src-mac**

## EtherChannel Config

- There are 3 methods of EtherChannel configuration on Cisco switches
  - PAgP (Port Aggregation Protocol)
    - Cisco proprietary protocol
    - Dynamically negotiates the creation/maintenance of the EtherChannel (like DTP for trunks)
  - LACP (Link Aggregation Control Protocol)
    - Industry standard protocol (IEEE 802.3ad)
    - Dynamically negotiates the creation/maintenance of the EtherChannel

- Static Channel
  - A protocol isn't used to determine if an EtherChannel should be formed
  - Interfaces are statically configured to form an EtherChannel
  - Should be avoided
- Up to 8 interfaces can be formed into a single EtherChannel (LACP allow 16 but 8 active, 8 standby)

```
ASW1(config)#interface range g0/0 - 3
ASW1(config-if-range)#channel-group 1 mode ?
  active   Enable LACP unconditionally
  auto     Enable PAgP only if a PAgP device is detected
  desirable  Enable PAgP unconditionally
  on       Enable Etherchannel only
  passive   Enable LACP only if a LACP device is detected

ASW1(config-if-range)#channel-group 1 mode desirable
Creating a port-channel interface Port-channel 1
```

auto + auto = no EtherChannel  
 desirable + auto = EtherChannel  
 desirable + desirable = EtherChannel

auto + auto = no EtherChannel  
 desirable + auto = EtherChannel  
 desirable + desirable = EtherChannel

- The channel-group number has to match for member interfaces on the same switch
- However, it doesn't have to match the channel group number on the other switch
- PAgP: "auto", "desirable"
- LACP: "active", "passive"
- Static: "on"
  - "On" mode only works with itself, will not form etherchannel with "desirable" or "active"

```
ASW1(config-if-range)#channel-protocol ?
  lacp  Prepare interface for LACP protocol
  pagg  Prepare interface for PAgP protocol

ASW1(config-if-range)#channel-protocol lacp
ASW1(config-if-range)#channel-group 1 mode desirable
Command rejected (Channel protocol mismatch for interface Gi0/0 in group 1): the interface can not be added to the channel group

% Range command terminated because it failed on GigabitEthernet0/0
ASW1(config-if-range)#channel-group 1 mode on
Command rejected (Channel protocol mismatch for interface Gi0/0 in group 1): the interface can not be added to the channel group

% Range command terminated because it failed on GigabitEthernet0/0
ASW1(config-if-range)#channel-group 1 mode active
Creating a port-channel interface Port-channel 1

ASW1(config-if-range)#[
```

- Not that useful, but good to know for exam

```

ASW1(config)#interface port-channel 1
ASW1(config-if)#switchport trunk encapsulation dot1q
ASW1(config-if)#switchport mode trunk
ASW1(config-if)#do show interfaces trunk

Port      Mode          Encapsulation  Status       Native vlan
Po1       on           802.1q        trunking    1

Port      Vlans allowed on trunk
Po1       1-4094

Port      Vlans allowed and active in management domain
Po1       1

Port      Vlans in spanning tree forwarding state and not pruned
Po1       none

```

- Members must have the same matching configs
  - Same duplex
  - Same speed
  - Same switchport mode (access/trunk)
  - Same allowed VLANs/native VLAN (for trunk ports)
- If an interface don't match, it will be excluded from the EtherChannel

```

ASW1#show etherchannel summary
Flags: D - down      P - bundled in port-channel
      I - stand-alone S - suspended
      H - Hot-standby (LACP only)
      R - Layer3      S - Layer2
      U - in use      N - not in use, no aggregation
      + - failed to allocate aggregator

      M - not in use, minimum links not met
      m - not in use, port not aggregated due to minimum links not met
      u - unsuitable for bundling
      w - waiting to be aggregated
      d - default port

      A - formed by Auto LAG

Number of channel-groups in use: 1
Number of aggregators: 1

```

Group	Port-channel	Protocol	Ports
1	Po1 (SU)	LACP	Gi0/0/(P) Gi0/1/(P) Gi0/2/(P)

```

ASW1(config)#interface po1
ASW1(config-if)#shutdown
ASW1(config-if)#do show etherchannel summary
Flags: D - down P - bundled in port-channel
      I - stand-alone s - suspended
      H - Hot-standby (LACP only)
      R - Layer3 S - Layer2
      U - in use N - not in use, no aggregation
      f - failed to allocate aggregator

      M - not in use, minimum links not met
      m - not in use, port not aggregated due to minimum links not met
      u - unsuitable for bundling
      w - waiting to be aggregated
      d - default port

      A - formed by Auto LAG

Number of channel-groups in use: 1
Number of aggregators: 1

Group Port-channel Protocol Ports
-----+-----+-----+
1     Po1(SD)      LACP   Gi0/0(D)  Gi0/1(D)  Gi0/2(D)
                  Gi0/3(D)

```

```

ASW1(config)#interface g0/0
ASW1(config-if)#switchport mode access
ASW1(config-if)#do show etherchannel summary
Flags: D - down P - bundled in port-channel
      I - stand-alone s - suspended
      H - Hot-standby (LACP only)
      R - Layer3 S - Layer2
      U - in use N - not in use, no aggregation
      f - failed to allocate aggregator

      M - not in use, minimum links not met
      m - not in use, port not aggregated due to minimum links not met
      u - unsuitable for bundling
      w - waiting to be aggregated
      d - default port

      A - formed by Auto LAG

Number of channel-groups in use: 1
Number of aggregators: 1

Group Port-channel Protocol Ports
-----+-----+-----+
1     Po1(SU)      LACP   Gi0/0(s)  Gi0/1(P)  Gi0/2(P)
                  Gi0/3(P)

```

```

ASW1#show etherchannel port-channel
  Channel-group listing:
  -----
  Group: 1
  -----
    Port-channels in the group:
  -----
  Port-channel: Po1      (Primary Aggregator)
  -----
  Age of the Port-channel = 0d:00h:36m:48s
  Logical slot/port = 16/0          Number of ports = 4
  HotStandBy port = null
  Port state       = Port-channel Ag-Inuse
  Protocol        = LACP
  Port security   = Disabled

  Ports in the Port-channel:
  Index  Load  Port      EC state      No of bits
  -----+-----+-----+-----+
  0     00   Gi0/0    Active           0
  0     00   Gi0/1    Active           0
  0     00   Gi0/2    Active           0
  0     00   Gi0/3    Active           0

  Time since last port bundled: 0d:00h:00m:02s  Gi0/0
  Time since last port Un-bundled: 0d:00h:08m:42s  Gi0/0

```

## Spanning Tree

```

ASW1#show spanning-tree

VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    32769
              Address     0c04.cf10.ea00
              This bridge is the root
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769 (priority 32768 sys-id-ext 1)
              Address     0c04.cf10.ea00
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time   300 sec

  Interface      Role Sts Cost      Prio.Nbr Type
  -----+-----+-----+-----+
  Po1          Desg FWD 3        128.65  Shr

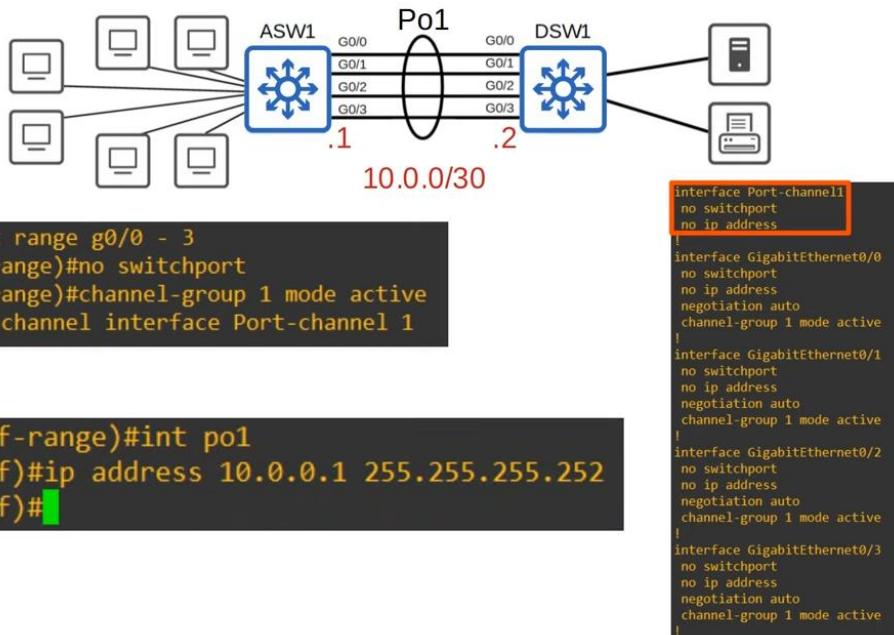
```

- Spanning tree treats the EtherChannel as a single interface

## Layer 3 EtherChannel

- Better since at layer 3, spanning tree is not an issue as broadcast storm won't happen
- Layer 2 loops will not occur
- Spanning tree won't be running

## Config



```
SW(config) port-channel load-balance mode  
#configures the EtherChannel load-balancing method on the switch

SW# show etherchannel load-balance  
#displays information about the load-balancing settings

SW(config-if)# channel-group number mode {desirable|auto|active|passive|on}  
#configures an interface to be part of an EtherChannel

SW# show etherchannel summary  
#displays a summary of EtherChannels on the switch

SW# show etherchannel port-channel  
#displays information about the virtual port-channel interfaces on the switch
```

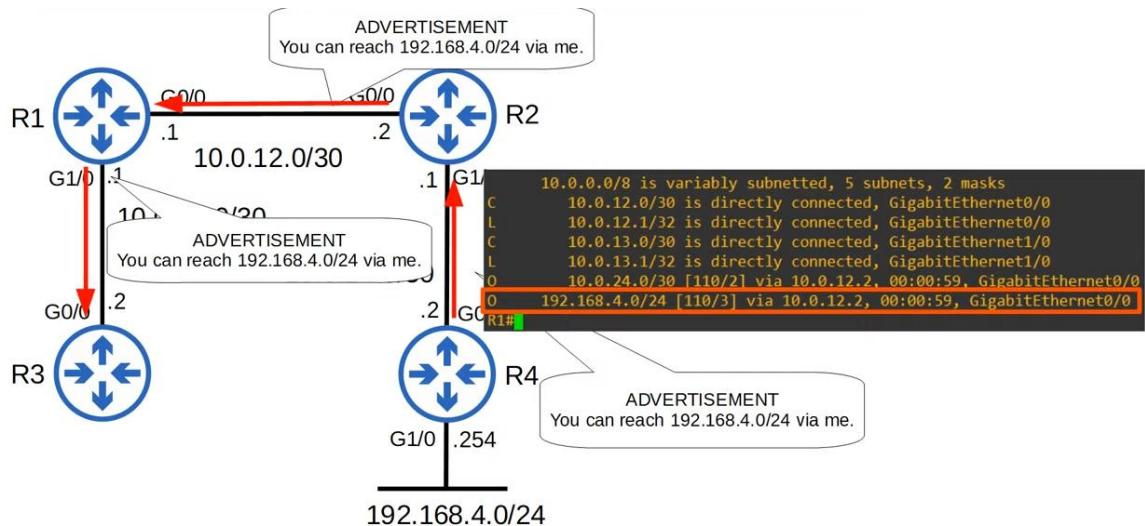
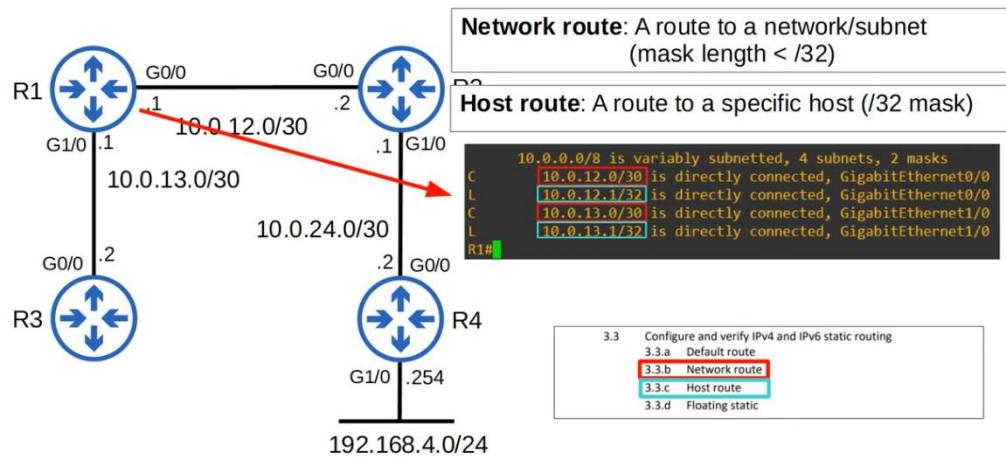
## Dynamic Routing

### Intro to Dynamic Routing

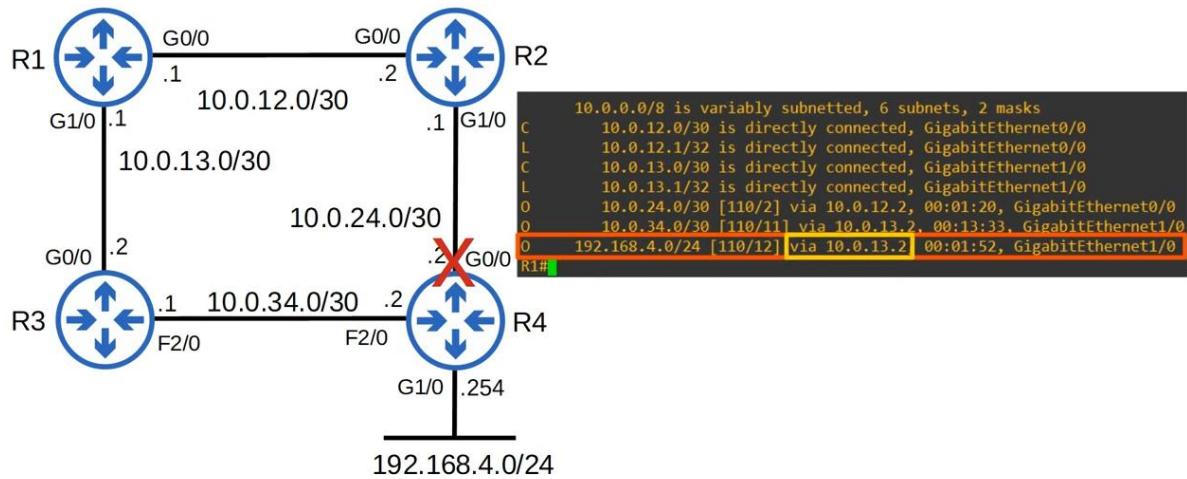
Things covered

- Intro to dynamic routing protocols
- Types of dynamic routing protocols
- Dynamic routing protocol metrics
- Administrative distance

## Dynamic Routing



- If one of the link fails, will automatically remove that route if no alternative route
  - If static route, the route will still remain

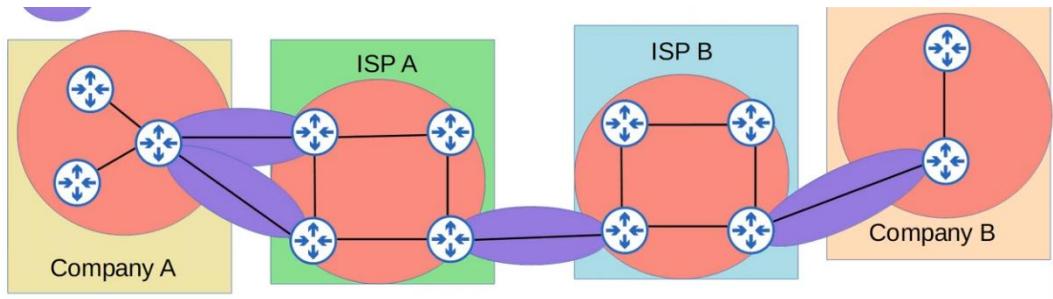


- The dynamic routing protocol will select the path with the lowest cost, that's why R1->R2->R4 selected first
  - The route only changed when R4 interface was down
- Key points
  - Routers can use dynamic routing protocols to advertise info about the routes they know to other routers
  - They form 'adjacencies' / 'neighbour relationships' / 'neighbourships' with adjacent routers to exchange info
  - If multiple routes to a destination are learned, the router determines which route is superior and adds it to the routing table
    - Uses 'metric' of the route to determine which is superior (lower metric = superior)

## Types of Dynamic Routing Protocols

- Can be divided into 2 categories
  - IGP (Interior Gateway Protocol)
  - EGP (Exterior Gateway Protocol)
- IGP
  - Used to share routes within a single autonomous system (AS), which is a single organization

- EGP
  - Used to share routes between different AS

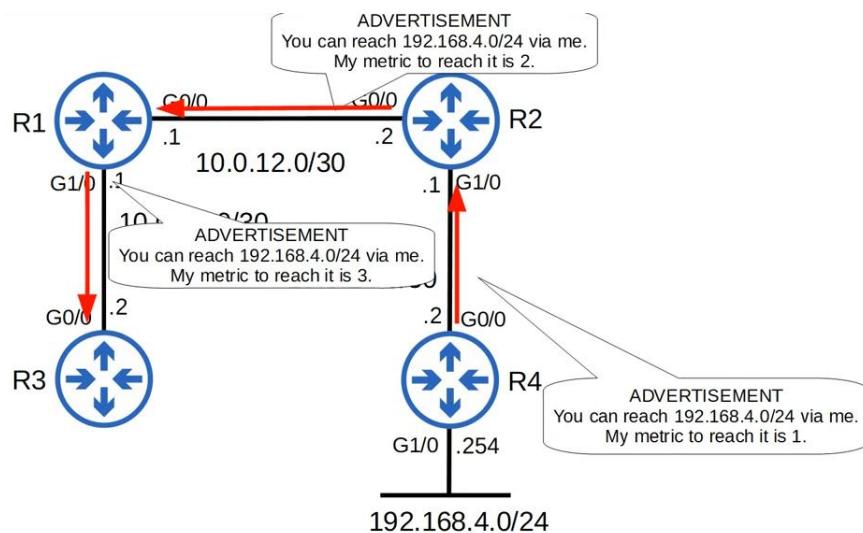


- IGP
  - Distance Vector
    - RIP (Routing Interface Protocol)
    - EIGRP (Enhanced Gateway Routing Protocol)
  - Link State
    - OSPF (Open Shortest Path First)
    - Intermediate System to Intermediate System (IS-IS)
- EGP
  - Path Vector
    - BGP (Border Gateway Protocol)
  - For EGP, BGP is the only one that is used by the whole world today
  - Distance vector, link state, path vector - algorithm type

### Distance Vector Routing Protocols

- Distance vector protocols were invented before link state protocols
- Early examples are RIPv1 and Cisco's proprietary protocol IGRP (which was updated to EIGRP)
- Distance vector protocols operate by sending the following to their directly connected neighbours
  - Their known destination networks

- Their metric to reach their known destination networks
- This method of sharing information is often called 'routing by rumor'
- This is because the router doesn't know about the network beyond its neighbours. It only knows the info that its neighbours tell it
- Called 'distance vector' because the routers only learn the 'distance' (metric) and 'vector' (direction, next-hop router) of each route

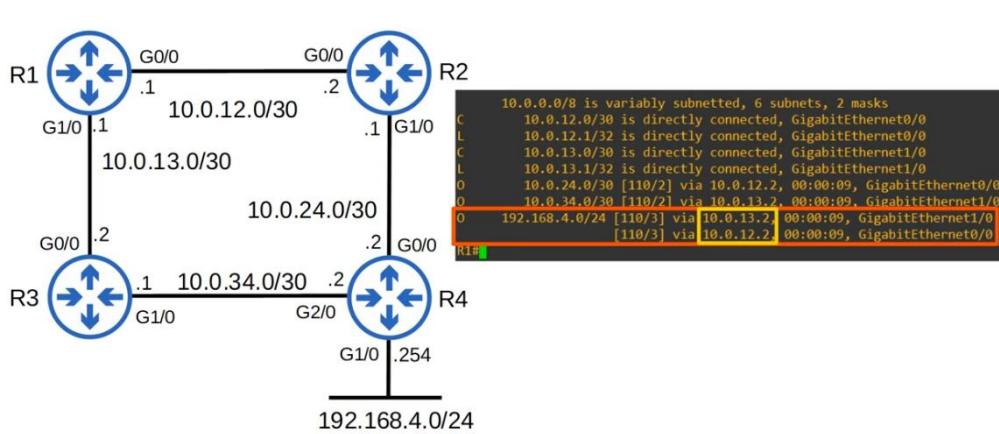


## Link State Routing Protocols

- When using a link state routing protocol, every router creates a 'connectivity map' of the network
- To allow this, each router advertises info about its interfaces (connected networks) to its neighbours. These advertisements are passed along to other routers, until all routers in the network develop the same map of the network
- Each router independently uses this map to calculate the best route to each destination
- Link state protocols use more CPU resource on the router, because more info is shared
- However, link state protocols tend to be faster in reacting to changes in the network than distance vector protocols

## Metrics

- A router table contains the best route to each destination network it knows about
- If there are multiple routes to a destination, uses metric to determine the best route
- Lower metric = better
- Each routing protocol uses a different metric to determine which route is the best
- If a router learns 2 or more routes via the same routing protocol to the same destination (same network address, same subnet mask) with the same metric, both will be added to the routing table. Traffic will be load-balanced over both routes
  - ECMP (Equal Cost Multi-Path)



- "[110/3]"
  - 3 - metric
  - 110 - administrative distance
- ECMP for static routes

```

R1(config)#ip route 192.168.4.0 255.255.255.0 10.0.12.2
R1(config)#ip route 192.168.4.0 255.255.255.0 10.0.13.2
R1(config)#do show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route, H - NHRP, l - LIS
      + - replicated route, % - next hop override
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.0.12.0/30 is directly connected, GigabitEthernet0/0
L        10.0.12.1/32 is directly connected, GigabitEthernet0/0
C        10.0.13.0/30 is directly connected, GigabitEthernet1/0
L        10.0.13.1/32 is directly connected, GigabitEthernet1/0
S        192.168.4.0/24 [1/0] via 10.0.13.2
                  [1/0] via 10.0.12.2
R1(config)#

```

IGP	Metric	Explanation
RIP	Hop count	Each router in the path counts as one 'hop'. The total metric is the total number of hops to the destination. <b>Links of all speeds are equal.</b>
EIGRP	Metric based on bandwidth & delay (by default)	Complex formula that can take into account many values. By default, the bandwidth of the <b>slowest link in the route</b> and the total delay of all links in the route are used.
OSPF	Cost	The cost of each link is calculated based on bandwidth. The total metric is the total cost of each link in the route.
IS-IS	Cost	The total metric is the total cost of each link in the route. The cost of each link is <b>not</b> automatically calculated by default. All links have a cost of 10 by default.

- Administrative Distance
  - In most cases a company will use a single IGP - usually OSPF or EIGRP
  - However, in some cases, they might use 2.
    - E.g. 2 companies connect their networks to share info
  - Metric is used to compare routes learned via the same routing protocol
  - Different routing protocol use totally different metrics, so they cannot be compared
    - E.g. For the route to a particular network, OSPF metric is 30 while EIGRP is 33280
    - Which route is better?

- The administrative distance (AD) is used to determine which routing protocol is preferred
- A lower AD is preferred, and indicates that the routing protocol is considered more 'trustworthy' (more likely to select good routes)

Route protocol/type	AD	Route protocol/type	AD
Directly connected	0	IS-IS	115
Static	1	RIP	120
External BGP (eBGP)	20	EIGRP (external)	170
EIGRP	90	Internal BGP (iBGP)	200
IGRP	100	Unusable route	255
OSPF	110		

- If the administrative distance is 255, the router does not believe the source of that route and does not install the route in the routing table
- You can change the AD of a routing protocol
  - R1(config-router)# **distance <AD>**
- You can change the AD of a static route
  - R1(config)# **ip route <ip addr> <netmask> <AD>**

### Floating Static Routes

- By changing the AD of a static route, can make it less preferred than routes learned by a dynamic routing protocol to the same destination (make sure the AD is higher than the dynamic routing protocol's AD)
- Called 'floating static routes'
- The route will be inactive (not in the routing table) unless the route learned by the dynamic routing protocol is removed
  - E.g. The router stops advertising it for some reason, or an interface failure causes an adjacency with a neighbour to be lost

# RIP & EIGRP

## Things covered

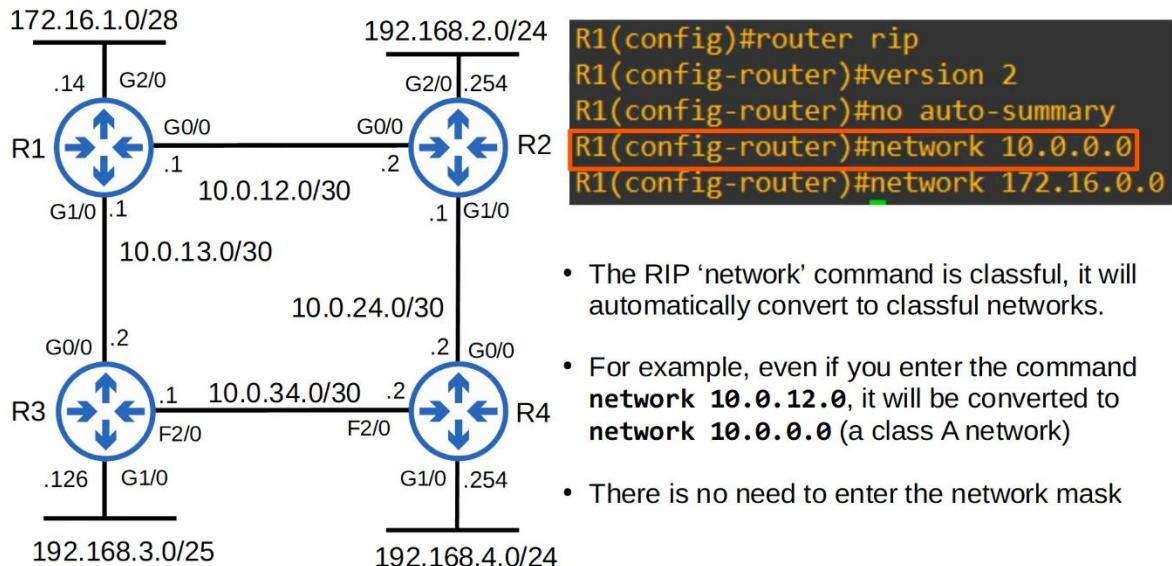
- Routing Interface Protocol
- Enhanced Gateway Routing Protocol

## Routing Interface Protocol (RIP)

- Industry standard, not propriety
- Distance vector IGP (uses routing-by-rumour logic to learn/share routes)
- Uses hop count as metric. 1 router = 1 hop (bandwidth is irrelevant)
- Max hop count = 15 (if >15, considered unreachable)
- Has 3 versions
  - RIPv1 and RIPv2 (IPv4)
  - RIPng (RIP Next Gen, for IPv6)
- Uses 2 message type
  - Request: Ask RIP-enabled neighbour routers to send their routing table
  - Response: Send the local router's routing table to neighbouring routers
- By default, RIP-enabled routers will share their routing table every 30s
- RIPv1
  - Only advertise classful addresses (Class A/B/C)
  - Don't support VLSM, CIDR
  - Don't include subnet mask information in advertisements (Response messages)
    - If IP address in class A range (1-127), it will assume /8
    - E.g. 10.1.1.0/24 -> 10.0.0.0 (assume /8 mask)
  - Message broadcast to 255.255.255.255

- RIPv2
  - Support VLSM and CIDR
  - Include subnet mask information in advertisements
  - Messages are multicast to 224.0.0.9
- Broadcast: messages are delivered to all devices on local network
- Multicast: messages are delivered only to devices that have joined that specific multicast group

## RIP Config



- The RIP ‘network’ command is classful, it will automatically convert to classful networks.
  - For example, even if you enter the command **network 10.0.12.0**, it will be converted to **network 10.0.0.0** (a class A network)
  - There is no need to enter the network mask
- "no auto-summary": disable default classful settings
  - "network" command tells the router to:
    - Look for interfaces with an IP address that is in the specified range
    - Active RIP on the interfaces that fall in the range
    - Form adjacencies with connected RIP neighbours
    - Advertise the network prefix of the interface (NOT the prefix in the network command)
  - OSPF and EIGRP network commands operate in the same way
  - "network 10.0.0.0"
    - Assumed to be 10.0.0.0/8

- R1 will look for any interfaces with an IP address that matches 10.0.0.0/8 (since /8, only first 8 bits need to match)
  - G0/0 and G1/0 match, so RIP is activated on their interfaces
  - R1 form adjacencies with R2 and R3
  - R1 advertises 10.0.12.0/30 and 10.0.13.0/30 (NOT 10.0.0.0/8) to its RIP neighbours
- "network" command does not tell the router which networks to advertise, it tells the router which interfaces to active RIP on, and then the router will advertise the network prefix of those interface
- "network 172.16.0.0"
  - Same as above
  - Although there are no RIP neighbours connected to G2/0, R1 will continuously send RIP advertisements out of G2/0
  - This is unnecessary traffic, so G2/0 should be configured as **passive interface**
- R1(config-router)# **passive-interface g2/0**
  - Tells router to stop sending advertisements on the interface
  - However, router will still advertise the network prefix of the interface (172.16.1.0/28) to its RIP neighbours (R2,R3)
  - Should always use this command
  - EIGRP and OSPF have the same passive interface functionality, using the same command
- R1(config-router)# **default-information originate**
  - To share default route

```

R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 28 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface      Send   Recv  Triggered RIP  Key-chain
    GigabitEthernet0/0   2      2
    GigabitEthernet1/0   2      2
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
    172.16.0.0
  Passive Interface(s):
    GigabitEthernet2/0
  Routing Information Sources:
    Gateway        Distance      Last Update
    10.0.12.2        120          00:00:21
    10.0.13.2        120          00:00:06
  Distance: (default is 120)

```

```

R1(config-router)#maximum-paths ?
<1-32>  Number of paths

R1(config-router)#maximum-paths 8

```

```

R1(config-router)#distance ?
<1-255>  Administrative distance

R1(config-router)#distance 85

```

- Distance
  - Administrative distance
  - Can be changed if you want RIP to be preferred over other methods
- Maximum path
  - Max number of paths if they have the same cost
  - Can be changed

## EIGRP

- Enhanced Interior Gateway Routing Protocol
- Was Cisco proprietary, but Cisco has now published it openly so other vendors can implement it on their equipment
- Considered 'advanced'/'hybrid' distance vector routing protocol
- Much faster than RIP in reacting to changes in the network
- Does not have the 15 hop count limit in RIP
- Sends messages using multicast address 224.0.0.10
- Is the only IGP that can perform unequal-cost load-balancing (by default it performs ECMP load-balancing over 4 paths like RIP)

## Configuration

```
R1(config)#router eigrp 1
R1(config-router)#no auto-summary
R1(config-router)#passive-interface g2/0
R1(config-router)#network 10.0.0.0
R1(config-router)#network 172.16.1.0 0.0.0.15
```

- "router eigrp 1"
  - 1 - Autonomous System number
  - Must match between routers, else they will not form adjacency and won't share info
- "network" command will assume classful address if mask not specified
- EIGRP uses a wildcard mask instead of a regular subnet mask
- Wildcard mask
  - All the 1 becomes 0
  - $255.255.255.0 = 0.0.0.255$
  - '0' in wildcard mask - must match
  - '1' in wildcard mask - don't need to match

```
R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "eigrp 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP-IPv4 Protocol for AS(1)
    Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
    NSF-aware route hold timer is 240
    Router-ID: 172.16.1.14
    Topology : 0 (base)
      Active Timer: 3 min
      Distance: internal 90 external 170
      Maximum path: 4
      Maximum hopcount 100
      Maximum metric variance 1

  Automatic Summarization: disabled
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
    172.16.1.0/28
  Passive Interface(s):
    GigabitEthernet2/0
  Routing Information Sources:
    Gateway          Distance      Last Update
    10.0.12.2        90           00:00:23
    10.0.13.2        90           00:00:23
[Distance: internal 90 external 170]
```

Router ID order of priority:

- 1) Manual configuration
- 2) Highest IP address on a loopback interface
- 3) Highest IP address on a physical interface

```
R1(config-router)#eigrp router-id ?
  A.B.C.D  EIGRP Router-ID in IP address format
R1(config-router)#eigrp router-id 1.1.1.1
```

```
Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
NSF-aware route hold timer is 240
Router-ID: 1.1.1.1
Topology : 0 (base)
```

```

R1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
      + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C        10.0.12.0/30 is directly connected, GigabitEthernet0/0
L        10.0.12.1/32 is directly connected, GigabitEthernet0/0
C        10.0.13.0/30 is directly connected, GigabitEthernet1/0
L        10.0.13.1/32 is directly connected, GigabitEthernet1/0
D        10.0.24.0/30 [90/3072] via 10.0.12.2, 00:11:09, GigabitEthernet0/0
D        10.0.34.0/30 [90/28416] via 10.0.13.2, 00:11:09, GigabitEthernet1/0
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.1.0/28 is directly connected, GigabitEthernet2/0
L        172.16.1.14/32 is directly connected, GigabitEthernet2/0
D        192.168.2.0/24 [90/3072] via 10.0.12.2, 00:11:09, GigabitEthernet0/0
      192.168.3.0/25 is subnetted, 1 subnets
D          192.168.3.0 [90/3072] via 10.0.13.2, 00:11:10, GigabitEthernet1/0
D          192.168.4.0/24 [90/3328] via 10.0.12.2, 00:11:09, GigabitEthernet0/0

```

## OSPF

### OSPF (Part 1)

Things covered

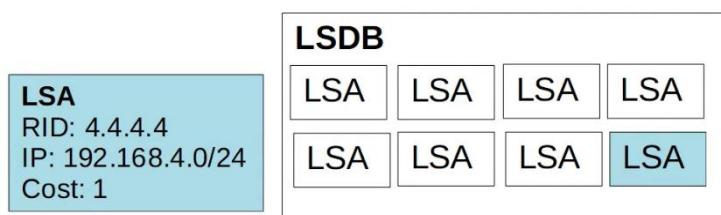
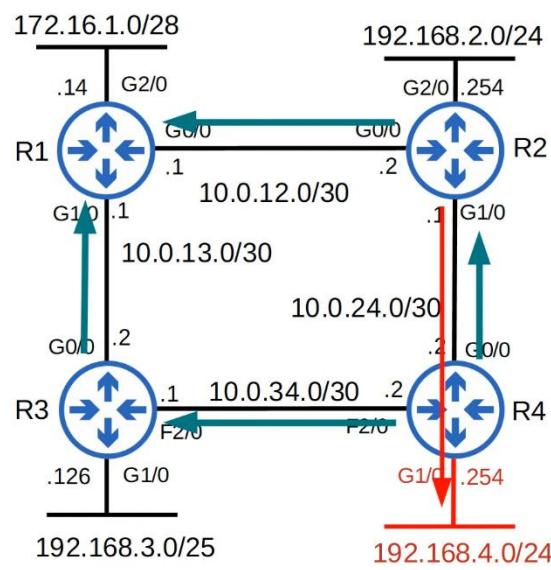
- Intro to OSPF
- OSPF Areas
- Basic OSPF Configurations

## OSPF

- Open Shortest Path First

- Uses Shortest Path First (Dijkstra) algorithm
- 3 versions
  - OSPFv1 (1989) - old, not used anymore
  - OSPFv2 (1998) - used for IPv4
  - OSPFv3 (2008) - used for IPv6 (can use for IPv4 also, but normally use v2)
- Routers store info about the network in LSAs (Link State Advertisements), which are organized in a structure called LSDB (Link State Database)
- Routers will flood LSAs until all routers in the OSPF area develop the same map of the network (LSDB)

## LSA Flooding



- OSPF is enabled on R4's G1/0 interface
- R4 creates an LSA to tell its neighbours about the network on G1/0
- The LSA is flooded throughout the network until all routers have received it

- This results in all routers sharing the same LSDB
- Each router then uses SPF algorithm to calculate its best route to 192.168.4.0/24
- Each LSA has a timer (default: 30 mins). The LSA will be flooded again after timer expires

## **OSPF**

- In OSPF, there are 3 main steps in the process of sharing LSAs and determining the best route to each destination in the network
  1. Become neighbours with other routers connected to the same segment
  2. Exchange LSAs with neighbour routers
  3. Calculate the best routes to each destination, and insert them in the routing table

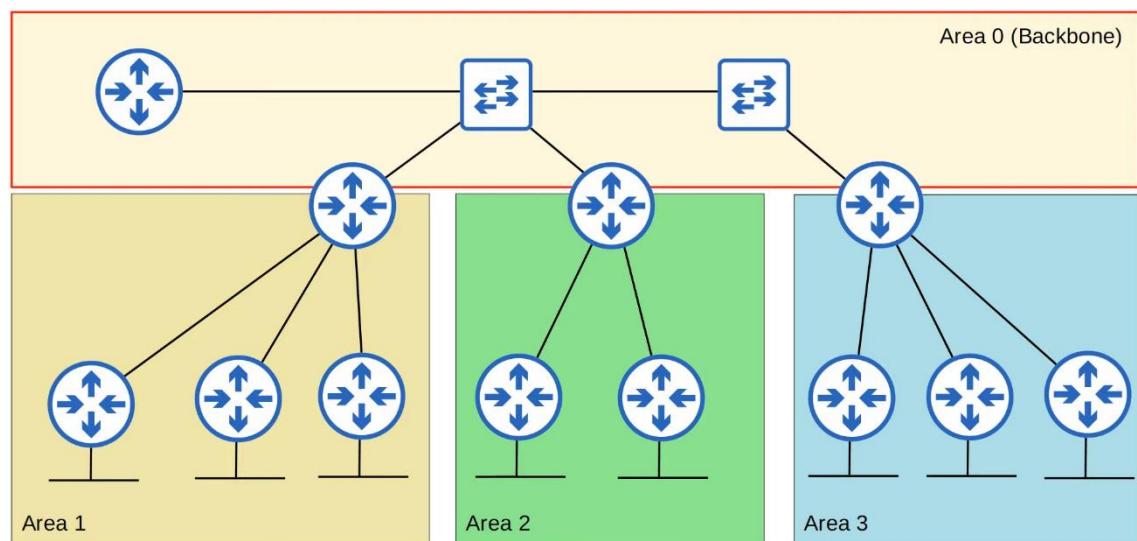
## **OSPF Areas**

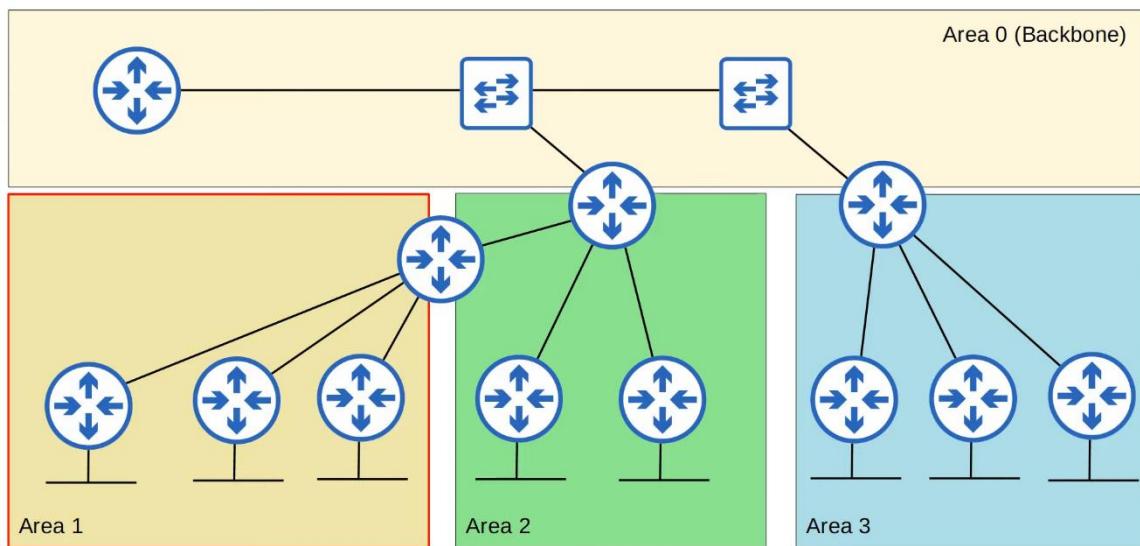
- OSPF uses areas to divide up the network
- Small networks can be single-area without any negative effects on performance
- In larger networks, a single-area design can have negative effects
  - SPF algorithm takes more time to calculate routes
  - SPF algorithm requires exponentially more processing power on the routers
  - Larger LSDB takes up more memory on the routers
  - Any small change in the network causes every router to flood LSAs and run the SPF algorithm again
- By dividing a large OSPF network into smaller areas, can avoid the negative effects

## **Terminologies**

- Area

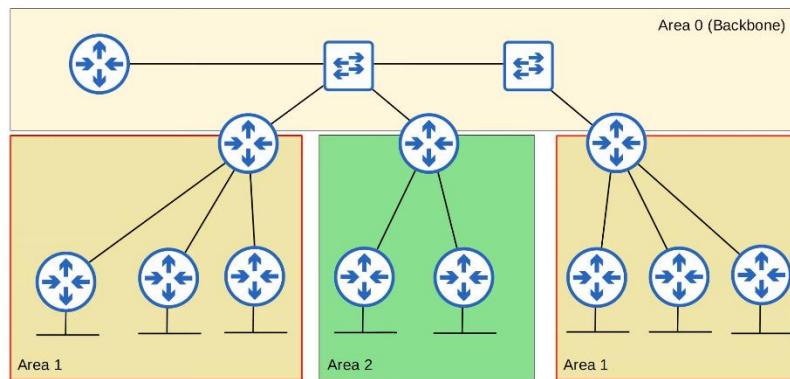
- Set of routers and links that share the same area
- Backbone area (area 0)
  - An area that all other areas must connect to
- Internal routers
  - All of the router's interfaces are in the same area
- Area Border Routers (ABRs)
  - Routers with interfaces in multiple areas
  - Maintain a separate LSDB for each area
  - Recommended to connect to a max of 2 areas. Can overburden the router if 3 areas
- Backbone Routers
  - Routers connected to the backbone area (Area 0)
- Intra-area route
  - A route to a destination inside the same OSPF area
- Inter-area route
  - A route to a destination in a different OSPF area

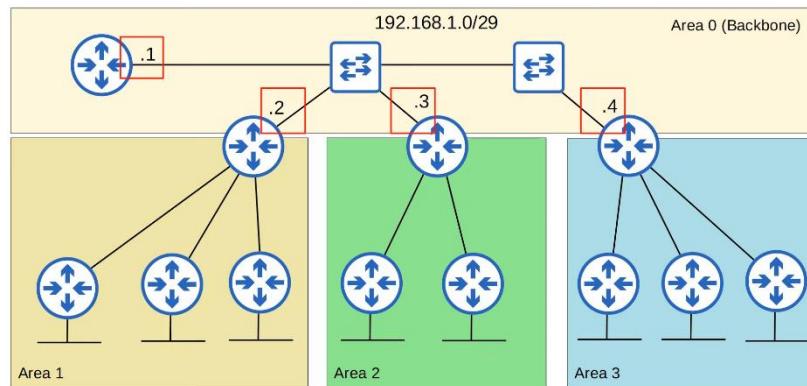
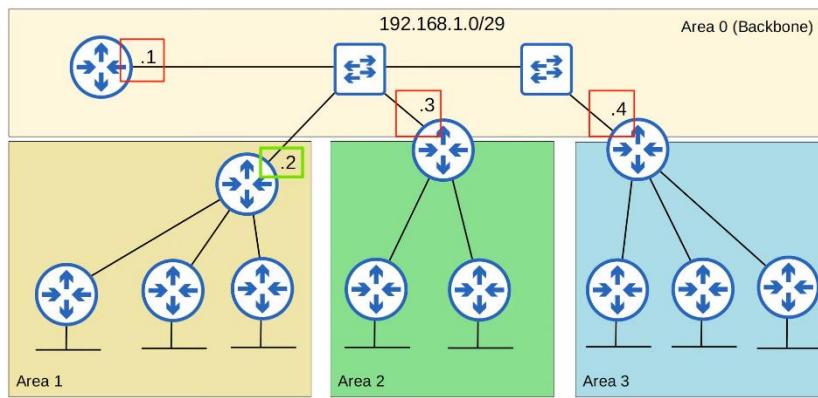




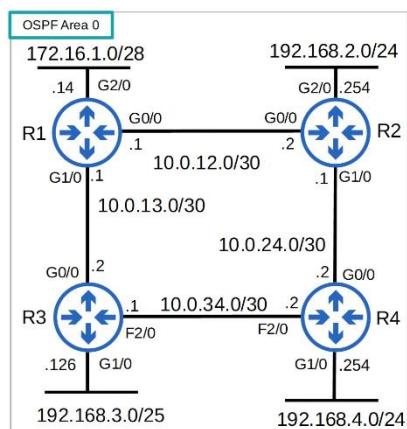
### OSPF Area Rules

- OSPF areas should be contiguous
- All OSPF areas must have at least 1 ABR connected to the backbone area
- OSPF interfaces in the same subnet must be in the same area





## Configurations



```

R1(config)#router ospf ?
<1-65535> Process ID

R1(config)#router ospf 1
R1(config-router)#network 10.0.12.0 0.0.0.3
% Incomplete command.

R1(config-router)#network 10.0.12.0 0.0.0.3 area 0
R1(config-router)#network 10.0.13.0 0.0.0.3 area 0
R1(config-router)#network 172.16.1.0 0.0.0.15 area 0
R1(config-router)#

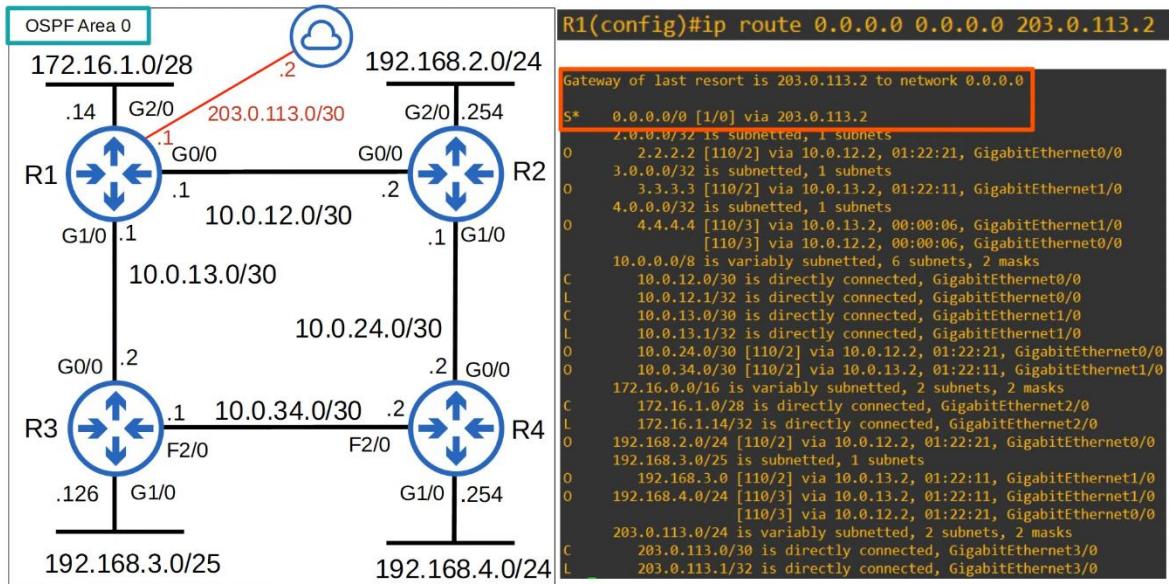
```

- "Process ID"
  - Locally significant
  - Will work if other routers have different ID
- "network"
  - Need to specify "area"
  - Tells OSPF to
    - Look for any interfaces with an IP address contained in the range specified in the "network" command
    - Activate OSPF in the specified area
    - The router will then try to become OSPF neighbours with other OSPF-activated neighbour routers
- For CCNA, only need to configure single-area OSPF (area 0)

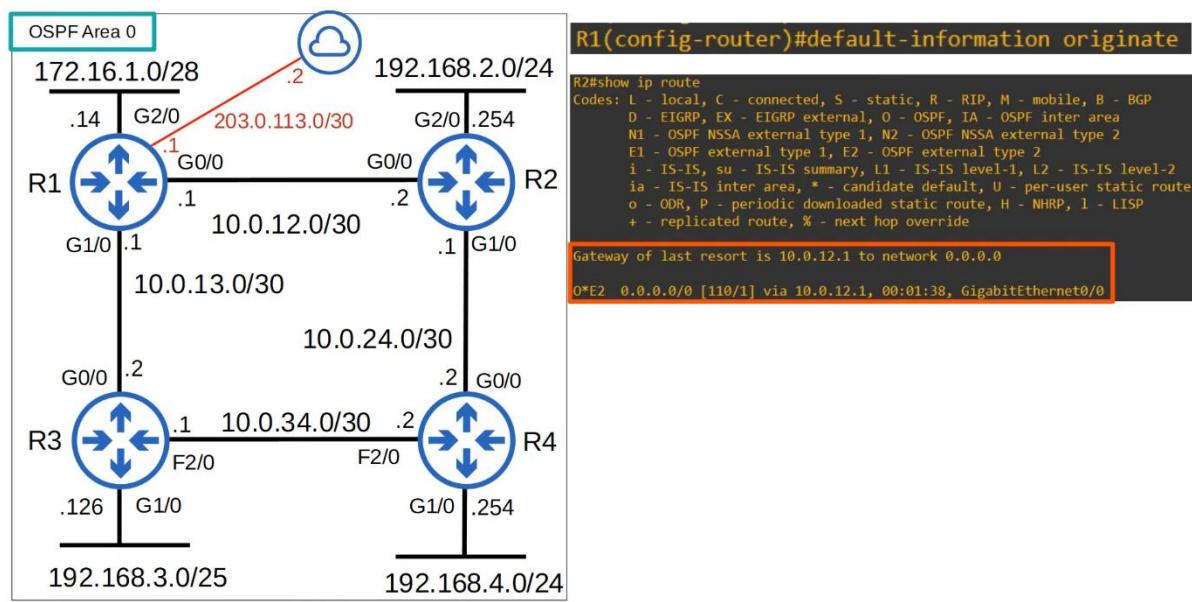
## Passive Interface

- R1(config-router)# passive-interface g2/0
  - Same as in RIP and EIGRP
  - Tells the router to stop sending out "hello" messages from the specified interface
  - However, the router will continue to send LSAs informing neighbours about the subnet configured on the interface
  - Should use this command for interfaces not connected to OSPF neighbours

## Default Route



### Advertise Default Route



**"show ip protocols"**

```
R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 172.16.1.14
    It is an autonomous system boundary router
  Redistributing External Routes from,
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.0.12.0 0.0.0.3 area 0
    10.0.13.0 0.0.0.3 area 0
    172.16.1.0 0.0.0.15 area 0
  Passive Interface(s):
    GigabitEthernet2/0
  Routing Information Sources:
    Gateway          Distance      Last Update
    4.4.4.4           110          00:00:08
    2.2.2.2           110          00:01:07
    3.3.3.3           110          00:01:07
    192.168.4.254    110          00:02:29
  Distance: (default is 110)
```

Router ID order of priority:

- 1) Manual configuration
- 2) Highest IP address on a loopback interface
- 3) Highest IP address on a physical interface

```
R1(config-router)#router-id ?
  A.B.C.D OSPF router-id in IP address format
R1(config-router)#router-id 1.1.1.1
% OSPF: Reload or use "clear ip ospf process" command, for this to take effect
```

```
R1#clear ip ospf process
Reset ALL OSPF processes? [no]: yes
R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 1.1.1.1
```

- Need additional command to change router id
  - Not recommended since will clear LSDB, but ok for CCNA
- Autonomous System Boundary Router (ASBR)
  - An OSPF router that connects the OSPF network to an external network
  - R1 is connected to the Internet
  - By using the "default-information originate" command, R1 becomes an ASBR
- R1(config-router)# **maximum-paths <no of paths>**
  - Number of paths allowed for ECMP
  - OSPF don't support unequal cost load balancing
- R1(config-router)# **distance <AD>**
  - Change the administrative distance for OSPF on the router

## OSPF (Part 2)

Things covered

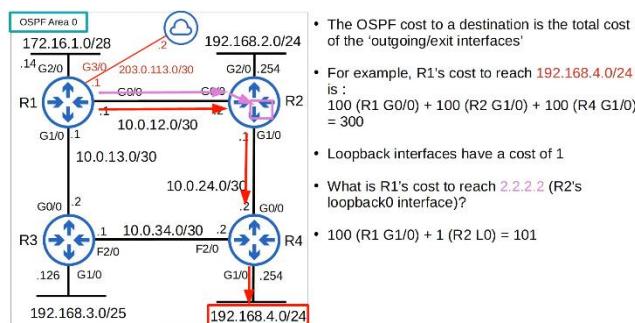
- OSPF metric (cost)
- Becoming OSPF neighbours
- More OSPF config

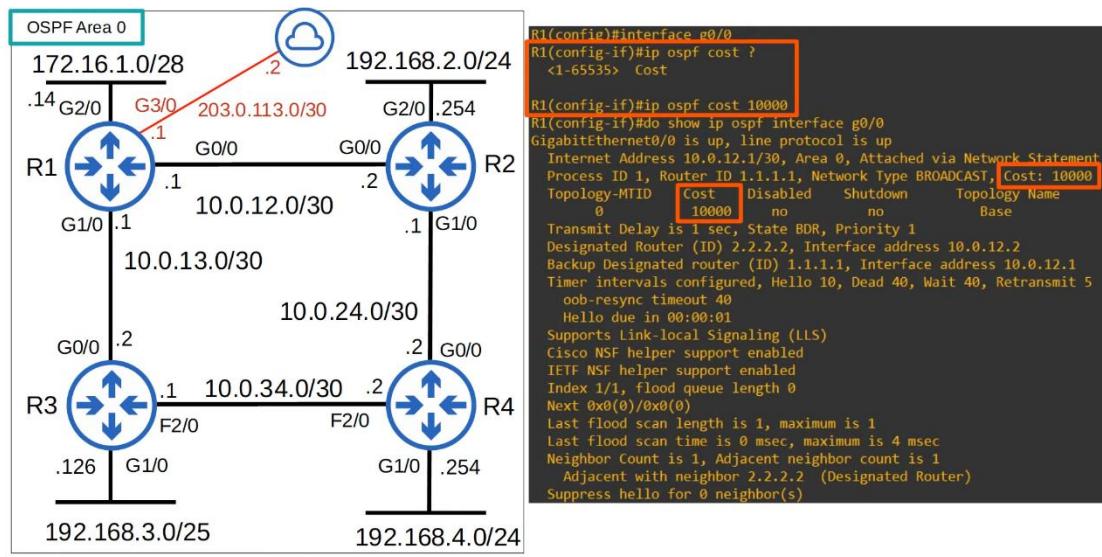
## OSPF Cost

- OSPF's metric is called cost
- Auto calculated based on bandwidth (speed) of the interface
- Calculated by dividing the a reference bandwidth value by the interface's bandwidth
- The default reference bandwidth is 100 mbps
  - Reference: 100 mbps / Interface: 10 mbps = cost of 10
  - Reference: 100 mbps / Interface: 100 mbps = cost of 1
  - Reference: 100 mbps / Interface: 1000 mbps = cost of 1
  - Reference: 100 mbps / Interface: 10000 mbps = cost of 1
- All values less than 1 will be converted to 1
- FastEthernet, Gigabit Ethernet, 10Gig Ethernet etc are equal and have a default cost of 1
- Can change the cost (and should do so)
  - R1(config-router)# **auto-cost reference-bandwidth <speed in megabits-per-second>**

```
R3(config-router)#auto-cost reference-bandwidth ?
<1-4294967>  The reference bandwidth in terms of Mbits per second
R3(config-router)#auto-cost reference-bandwidth 100000
% OSPF: Reference bandwidth is changed.
  Please ensure reference bandwidth is consistent across all routers.
R3(config-router)#[
```

- $100000/100 = \text{cost of 1000}$  for FastEthernet
- $100000/1000 = \text{cost of 100}$  for Gig Ethernet
- Should configure a reference bandwidth that is greater than the fastest link in your networks (allow for future upgrades)
- Should configure the same reference bandwidth for all OSPF routers





- One more option to change the OSPF cost of an interface is to change the bandwidth of the interface with "**bandwidth**" command
- The formula to calculate OSPF cost is "**reference bandwidth**" / "**interface bandwidth**"
- Although the bandwidth matches the interface speed by default, changing the interface bandwidth doesn't actually change the speed at which the interface operates
- Bandwidth is just used to calculate the OSPF cost, EIGRP metric etc
- To change actual speed, use "**speed**"
- Because bandwidth is used in other calculations, not recommended to change
- Recommended to change the reference bandwidth, and then use "**ip ospf cost**" command to change the cost of the individual interfaces you want

```

R1(config-if)#bandwidth ?
<1-10000000> Bandwidth in kilobits
inherit Specify how bandwidth is inherited
qos-reference Reference bandwidth for QoS test
receive Specify receive-side bandwidth

```

- The command above is to change the actual bandwidth
- In Kbps
- Not recommended

- 3 ways to modify OSPF cost
  1. Change reference bandwidth

R1(config-router)# **auto-cost reference-bandwidth <bandwidth Mbps>**

2. Manual config

R1(config-int)# **ip ospf cost <cost>**

3. Change the interface bandwidth

R1(config-int)# **bandwidth <bandwidth Kbps>**

- "show ip ospf int br"
  - o Check the cost of the interface

## OSPF Neighbours

- Making sure that routers successfully become OSPF neighbours is the main task in configuring

and troubleshooting OSPF.

- Once routers become neighbours, they automatically do the work of sharing network

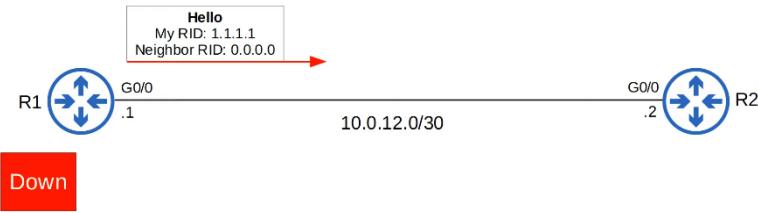
information, calculating routes, etc.

- When OSPF is activated on an interface, the router starts sending OSPF hello messages out

of the interface at regular intervals (determined by the hello timer). These are used to introduce the router to potential OSPF neighbours.

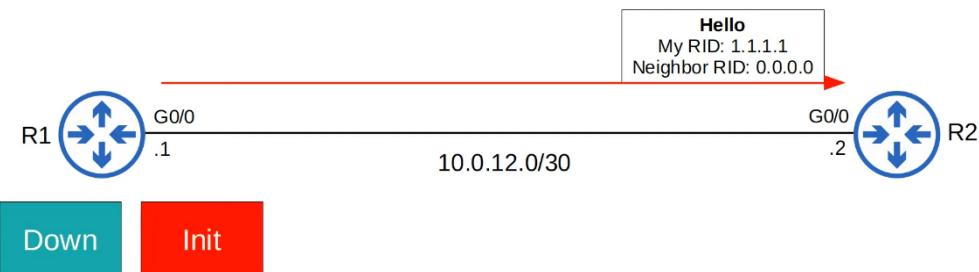
- The default hello timer is 10 seconds on an Ethernet connection.
- Hello messages are multicast to 224.0.0.5 (multicast address for all OSPF routers)
- OSPF messages are encapsulated in an IP header, with a value of 89 in the Protocol field.

## Down State



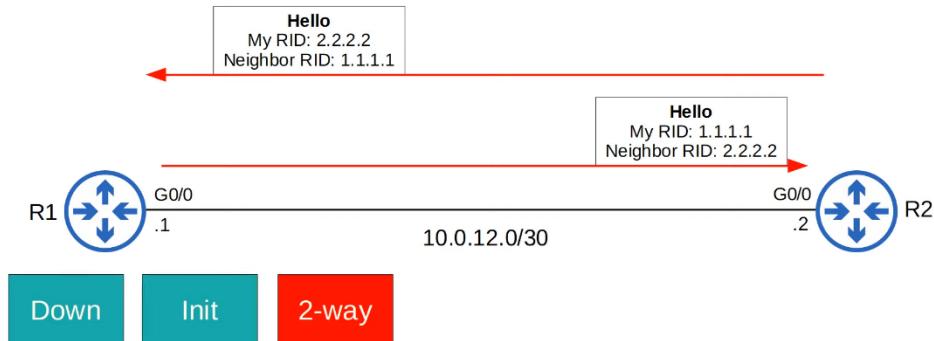
- OSPF is activated on R1's GO/O interface.
- It sends an OSPF hello message to 224.0.0.5.
- It doesn't know about any OSPF neighbours yet, so the current neighbour state is Down.

## Init State



- When R2 receives the Hello packet, it will add an entry for R1 to its OSPF neighbour table.
- In R2's neighbour table, the relationship with R1 is now in the Init state.
- Init state = Hello packet received, but own router ID is not in the Hello packet

## 2-way State



- R2 will send a Hello packet containing the RID of both routers.

R1 will insert R2 into its OSPF neighbour table in the 2-way state.

- R1 will send another Hello message, this time containing R2's RID.
- Now both routers are in the 2-way state.
- The 2-way state means the router has received a Hello packet with its own RID in it.
- If both routers reach the 2-way state, it means that all of the conditions have been met for them to become OSPF neighbours. They are now ready to share LSAs to build a common LSDB.
- In some network types, a DR (Designated Router) and BDR (Backup Designated Router) will be elected at this point.(Day 28)

### Exstart State



- The two routers will now prepare to exchange information about their LSDB.
- Before that, they have to choose which one will start the exchange.
- They do this in the Exstart state.
- The router with the higher RID will become the Master and initiate the exchange. The router with the lower RID will become the Slave.

with the lower RID will become the Slave.

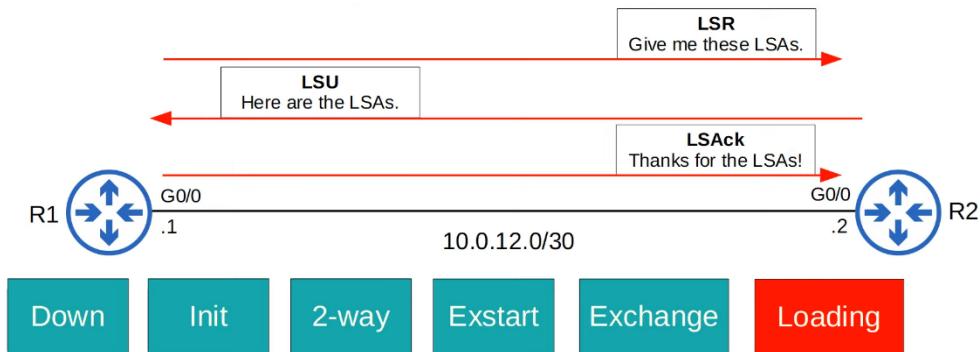
- To decide the Master and Slave, they exchange DBD (Database Description) packets.

## Exchange State



- In the Exchange state, the routers exchange DBDs which contain a list of the LSAs in their LSDB.
- These DBDs do not include detailed information about the LSAs, just basic information.
- The routers compare the information in the DBD they received to the information in their own LSDB to determine which LSAs they must receive from their neighbour.

## Loading State



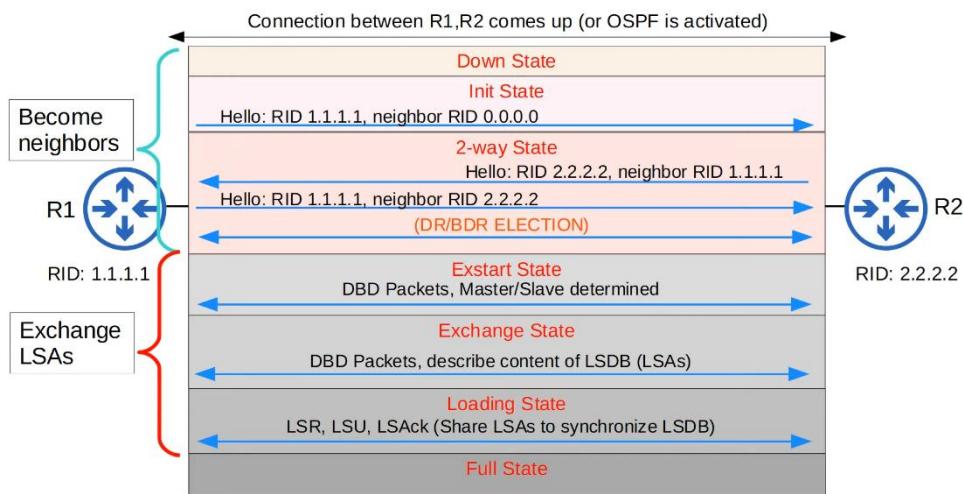
- In the Loading state, routers send Link State Request (LSR) messages to request that their neighbours send them any LSAs they don't have.
- LSAs are sent in Link State Update (LSU) messages.
- The routers send LSAck messages to acknowledge that they received the LSAs.

## Full State



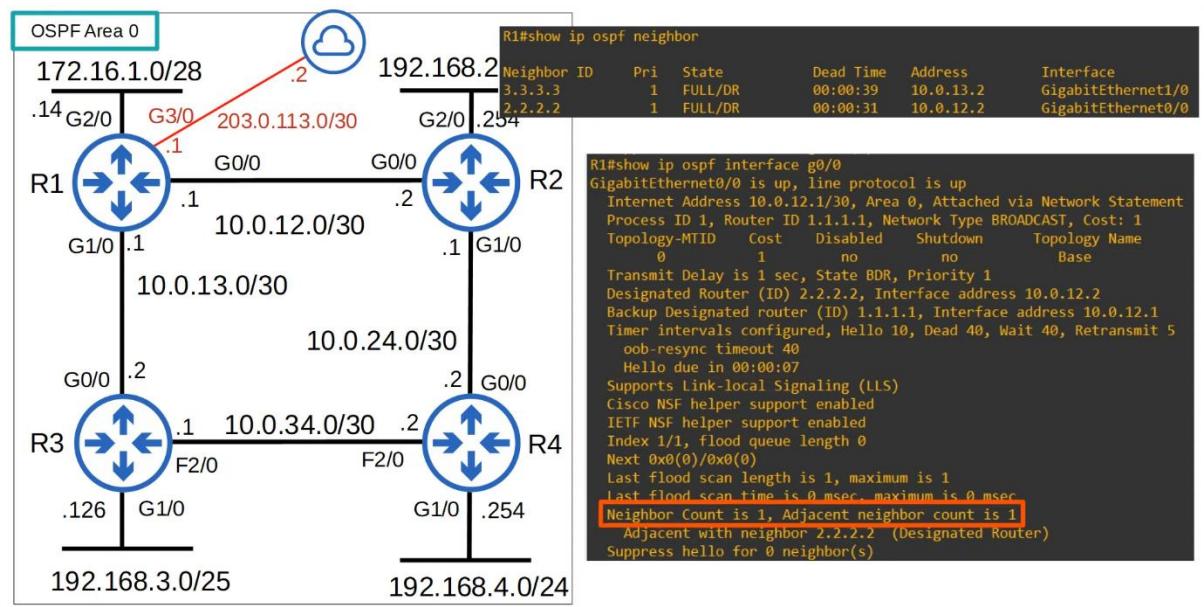
- In the Full state, the routers have a full OSPF adjacency and identical LSDBs.
- They continue to send and listen for Hello packets (every 10 seconds by default) to maintain the neighbour adjacency.
- Every time a Hello packet is received, the 'Dead' timer (40 seconds by default) is reset.
- If the Dead timer counts down to 0 and no Hello message is received, the neighbour is removed.
- The routers will continue to share LSAs as the network changes to make sure each router has complete and accurate map of the network (LSDB).

## Summary



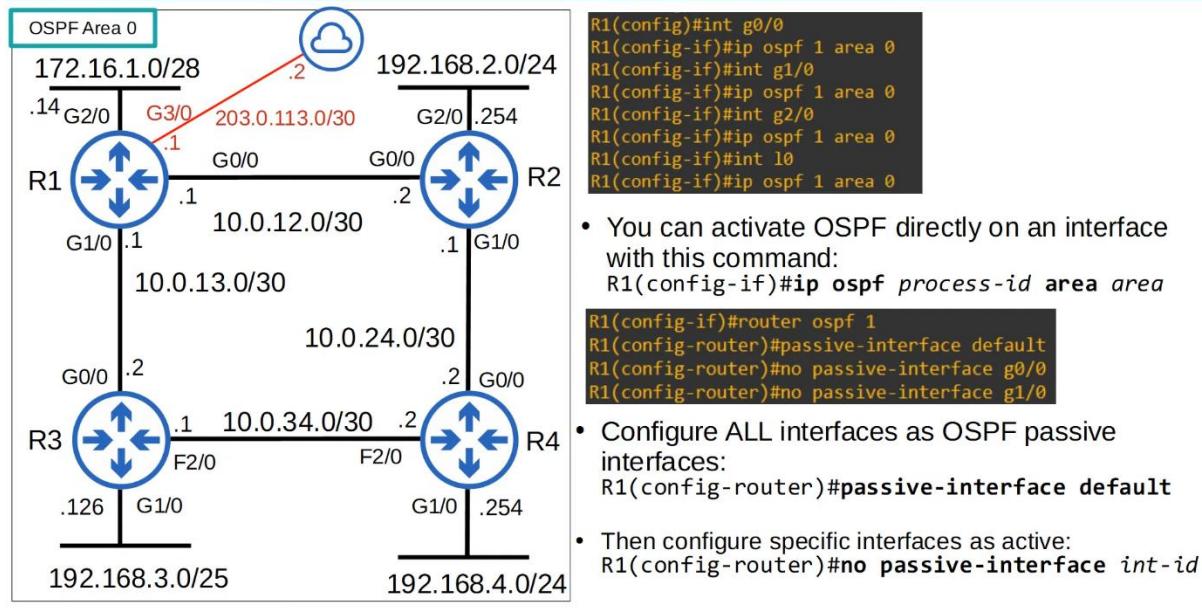
Type	Name	Purpose
1	<b>Hello</b>	Neighbor discovery and maintenance.
2	<b>Database Description (DBD)</b>	Summary of the LSDB of the router. Used to check if the LSDB of each router is the same.
3	<b>Link-State Request (LSR)</b>	Requests specific LSAs from the neighbor.
4	<b>Link-State Update (LSU)</b>	Sends specific LSAs to the neighbor.
5	<b>Link-State Acknowledgement (LSAck)</b>	Used to acknowledge that the router received a message.

## OSPF Neighbours



## Configurations

- Configure OSPF directly on interface



- If configure directly on interface, there will be some differences
  - "Routing for networks" will be empty
  - "Routing on Interfaces Configured Explicitly" will contain the interfaces
  - The rest will be the same

```

R1#show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 1.1.1.1
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    Routing on Interfaces Configured Explicitly (Area 0):
      Loopback0
      GigabitEthernet1/0
      GigabitEthernet0/0
      GigabitEthernet2/0
  Passive Interface(s):
    Ethernet0/0
    GigabitEthernet2/0
    GigabitEthernet3/0
    Loopback0
    VoIP-Null0
  Routing Information Sources:
    Gateway          Distance      Last Update
    2.2.2.2           110          00:09:53
    Gateway          Distance      Last Update
    3.3.3.3           110          00:09:54
    4.4.4.4           110          00:09:54
  Distance: (default is 110)

```

## Overall Summary

### OSPF metric (cost)

- Reference bandwidth / interface bandwidth = cost (values less than 1 are converted to 1)
- Default reference bandwidth = 100 mbps
- Modify the reference bandwidth:

R1 (config-router)# **auto-cost reference-bandwidth <megabits-per-second>**

- Manually configure the cost of an interface:

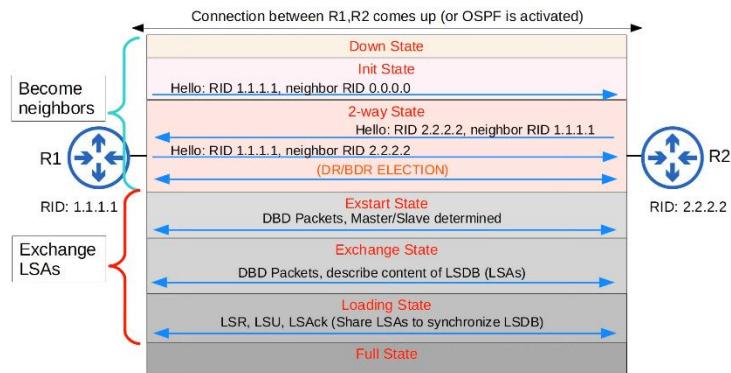
R1 (config-if)# **ip ospf cost <cost>**

- Modify the interface bandwidth:

R1 (config-if)# **bandwidth <kilobits-per-second>**

- Total cost of outgoing interfaces = metric of the route

## OSPF Neighbours



## More OSPF Configuration

- Activate OSPF directly on an interface:

R1 (config-if)# **ip ospf <process-id> area <area-id>**

- Configure all interfaces as passive interfaces by default:

R1 (config-router)# **passive-interface default**

## Message Type

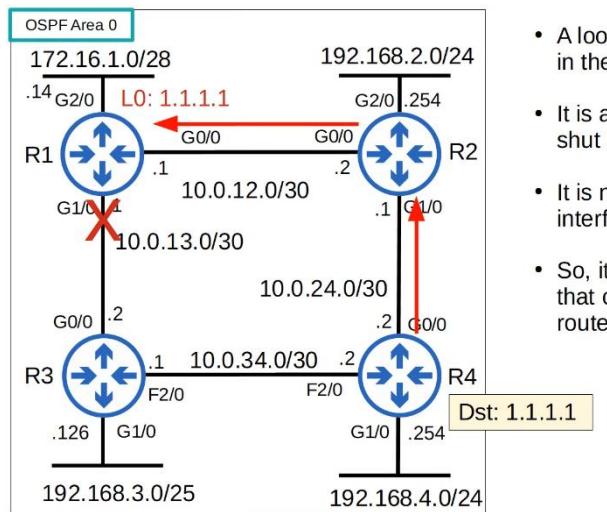
- Hello: 1
- DBD: 2
- LSU: 3

## Topics covered

- OSPF network types
- OSPF neighbour/adjacencies requirements
- OSPF LSA types

## Loopback Interfaces

- A virtual interface in the router
- Always up/up (unless manually shutdown)
- Not dependent on a physical interface
  - Won't fail unless router fail
- So, it provides a consistent IP address that can be used to reach/identify the router



- A loo|  
in the
- It is a  
shut i
- It is n  
interfa
- So, it  
that c  
route

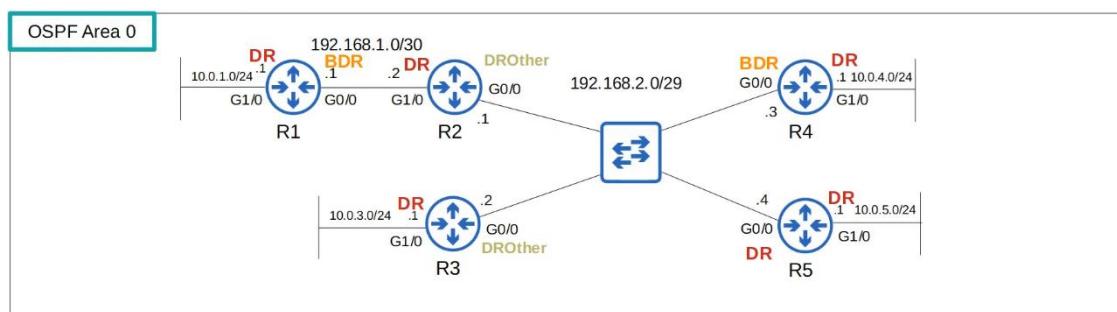
## OSPF Network Types

- Refers to the type of connection btw OSPF neighbours (Ethernet etc)

- There are 3 main OSPF network types
  - Broadcast
    - Enabled by default on
      - **Ethernet** interfaces
      - **FDDI (Fiber Distributed Data Interfaces)** interfaces
  - Point-to-Point
    - Enabled by default on
      - **PPP(Point-to-point)** interfaces
      - **HDLC (High-Level Data Link Control)** interfaces
  - Non-broadcast
    - Enabled by default on
      - **Frame Relay** interfaces
      - **X.25** interfaces

## Broadcast Network Type

- Enabled on Ethernet and FDDI interfaces by default
- Routers dynamically discover neighbours by sending/listening for OSPF hello messages using multicast address 224.0.0.5
  - Non-broadcast need to manually configure neighbours (not in CCNA)
- A DR(designated router) and BDR (backup designated router) must be elected on each subnet
  - Only DR elected if there are no OSPF neighbours (e.g. R1 G1/0 interface)
- Routers that are not DR or BDR are DROther



- DR/DBR election
  1. Highest OSPF interface priority
  2. Highest OSPF router ID
- 1st place become DR, 2nd place become BDR
- By default, OSPF interface priority is 1 on all interfaces

```
R2#show ip ospf int g0/0
GigabitEthernet0/0 is up, line protocol is up
  Internet Address 192.168.2.1/29, Area 0, Attached via Network Statement
  Process ID 1, Router ID 2.2.2.2, Network Type BROADCAST, Cost: 1
  Topology-MTID    Cost    Disabled    Shutdown    Topology Name
    0            1        no          no          Base
  Transmit Delay is 1 sec, State DROTHER, Priority 1
  Designated Router (ID) 5.5.5.5, Interface address 192.168.2.4
  Backup Designated router (ID) 4.4.4.4, Interface address 192.168.2.3
```

```
R2(config)#int g0/0
R2(config-if)#ip ospf priority ?
<0-255> Priority
```

```
R2(config-if)#ip ospf priority 255
```

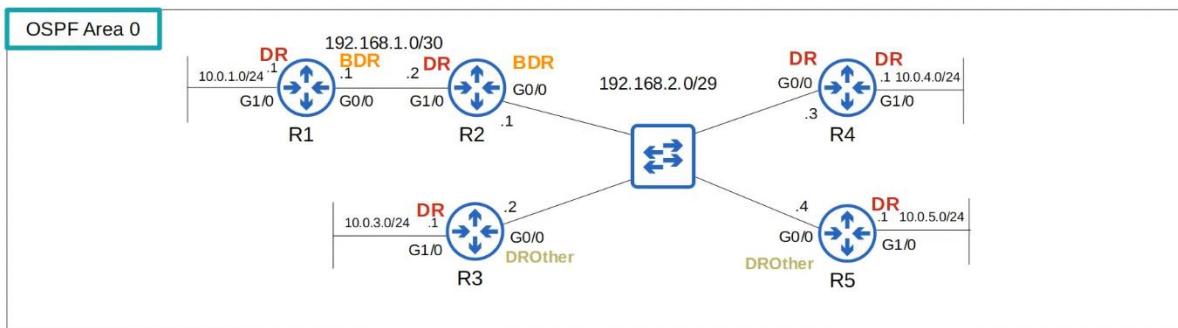
- If set to 0, router cannot become DR/BDR for the subnet
- However, R2 will still remain DROther
  - DR/DBR selection is non-pre-emptive
  - Once they are selected, will not change until OSPF is reset (interface fail/shutdown)

```
R5#clear ip ospf process
Reset ALL OSPF processes? [no]: yes
R5#
*Aug 22 04:25:05.307: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: Interface down or detached
*Aug 22 04:25:05.311: %OSPF-5-ADJCHG: Process 1, Nbr 3.3.3.3 on GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: Interface down or detached
*Aug 22 04:25:05.311: %OSPF-5-ADJCHG: Process 1, Nbr 4.4.4.4 on GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: Interface down or detached
R5#
*Aug 22 04:25:13.903: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on GigabitEthernet0/0 from LOADING to FULL, Loading Done
*Aug 22 04:25:13.907: %OSPF-5-ADJCHG: Process 1, Nbr 4.4.4.4 on GigabitEthernet0/0 from LOADING to FULL, Loading Done
R5#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
2.2.2.2	255	FULL/BDR	00:00:37	192.168.2.1	GigabitEthernet0/0
3.3.3.3	1	2WAY/DROTHER	00:00:37	192.168.2.2	GigabitEthernet0/0
4.4.4.4	1	FULL/DR	00:00:39	192.168.2.3	GigabitEthernet0/0

- R4 became the DR, not R2
  - R2 became the new BDR
  - When the DR goes down, the BDR becomes the new DR

- Then, an election is held for the new BDR
  - R3 is a DROther, and is stable in the 2-way state
    - Only neighbour state btw DROther and DR/BDR will be Full
    - If btw DROther and DROther, it will be 2-way
- In the broadcast network type, routers will only form a full OSPF adjacency with the DR/BDR of the segment
- Therefore, routers will only exchange LSAs with DR/BDR
  - DROthers will not exchange LSAs with each other
- All routers will still have the same LSDB, but this reduces the amount of LSAs flooding in the network
- Messages to the DR/BDR are multicast using address 224.0.0.6



R3#show ip ospf interface brief							
Interface	PID	Area	IP Address/Mask	Cost	State	Nbrs	F/C
Gi0/0	1	0	192.168.2.2/29	1	DROTH	2/3	
Gi1/0	1	0	10.0.3.1/24	1	DR	0/0	

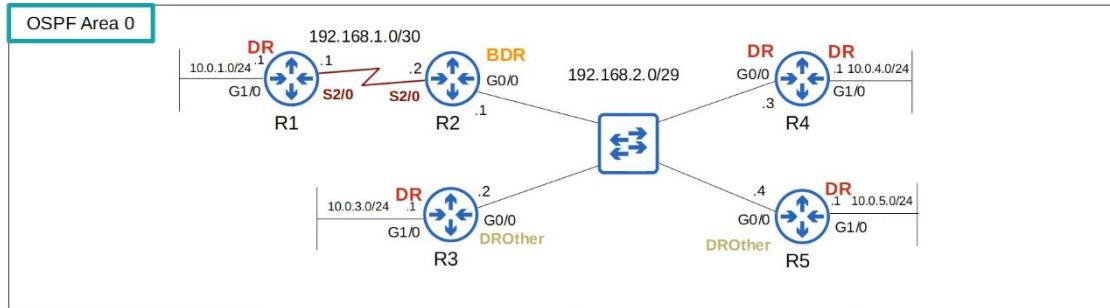
```

R3#show ip ospf interface g0/0
GigabitEthernet0/0 is up, line protocol is up
  Internet Address 192.168.2.2/29, Area 0, Attached via Network Statement
  Process ID 1, Router ID 3.3.3.3, Network Type BROADCAST, Cost: 1
  Topology-MTID    Cost    Disabled    Shutdown    Topology Name
    0            1        no        no        Base
  Transmit Delay is 1 sec, State DROTHER, Priority 1
  Designated Router (ID) 4.4.4.4, Interface address 192.168.2.3
  Backup Designated router (ID) 2.2.2.2, Interface address 192.168.2.1
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:09
  Supports Link-local Signaling (LLS)
  Cisco NSF helper support enabled
  IETF NSF helper support enabled
  Index 2/2, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 0, maximum is 1
  Last flood scan time is 0 msec, maximum is 4 msec
  Neighbor Count is 3, Adjacent neighbor count is 2
    Adjacent with neighbor 2.2.2.2 (Backup Designated Router)
    Adjacent with neighbor 4.4.4.4 (Designated Router)
  Suppress hello for 0 neighbor(s)

```

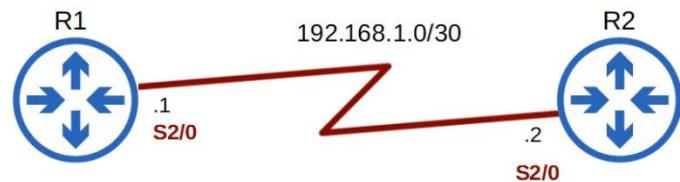
- R3 has 2 neighbours in 'Full' adjacency
- R3 has a neighbour count of 3

### Point-to-point Network Type



- Enabled on serial interfaces using PPP or HDLC encapsulation by default
- Routers dynamically discover neighbours by sending/listening for OSPF Hello messages using multicast address 224.0.0.5
- A DR and DBR is NOT elected
- These encapsulations are used for 'point-to-point' connections
- Therefore, there is no point in electing DR and BDR
- The 2 routers will form a Full adjacency with each other

## Serial Interfaces



```
R1(config)#interface s2/0
R1(config-if)#clock rate ?
With the exception of the following standard values not subject to rounding,
1200 2400 4800 9600 14400 19200 28800 38400
56000 64000 128000 2015232

accepted clockrates will be bestfitted (rounded) to the nearest value
supportable by the hardware.

<246-8064000>    DCE clock rate (bits per second)

R1(config-if)#clock rate 64000
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#no shut
```

- One side of a serial connection functions as DCE (Data Communications Equipment)
- The other side functions as DTE (Data Terminal Equipment)
- The DCE side needs to configure the clock rate (speed) of the connection
- Note
  - Ethernet: "speed"
  - Serial: "clock rate"

```
R1#show interface s2/0
Serial2/0 is up, line protocol is up
  Hardware is M4T
  Internet address is 192.168.1.1/24
  MTU 1500 bytes, BW 1544 Kbit/sec, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 16, loopback not set
```

- The default encapsulation on a serial interface is HDLC
  - Actually cHDLC (Cisco HDLC)

**cHDLC frame structure** [edit]

The following table describes the structure of a cHDLC frame on the wire. [citation needed]

Address	Control	Protocol Code	Information	Frame Check Sequence (FCS)	Flag
8 bits	8 bits	16 bits	Variable length, 0 or more bits, in multiples of 8	16 bits	8 bits

- The Address field is used to specify the type of packet contained in the cHDLC frame; 0x0F for Unicast and 0x8F for Broadcast packets.
- The Control field is always set to zero (0x00).
- The Protocol Code field is used to specify the protocol type encapsulated within the cHDLC frame (e.g. 0x0800 for Internet Protocol).

```
R1(config)#int s2/0
R1(config-if)#encapsulation ppp
R1(config-if)#do show interface s2/0
Serial2/0 is up, line protocol is up
  Hardware is M4T
  Internet address is 192.168.1.1/24
  MTU 1500 bytes, BW 1544 Kbit/sec, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, LCP Open
```

- If you change the encapsulation, it must match on both ends or the interface will go down

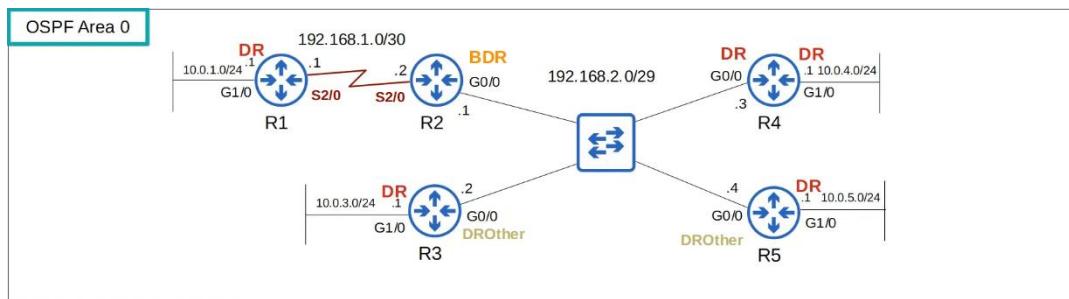
```
R1#show running-config interface s2/0
Building configuration...
!
Current configuration : 126 bytes
!
interface Serial2/0
  ip address 192.168.1.1 255.255.255.0
  encapsulation ppp
  serial restart-delay 0
  clock rate 64000
end
```

```
R2#show running-config interface s2/0
Building configuration...
!
Current configuration : 110 bytes
!
interface Serial2/0
  ip address 192.168.1.2 255.255.255.252
  encapsulation ppp
  serial restart-delay 0
end
```

## Summary (Serial Interfaces)

- Default encapsulation is HDLC
- Can configure encapsulation as PPP
  - R1(config)# **encapsulation ppp**
- One side is DCE and other side is DTE
- Identify which side is DCE/DTE
  - R1# **show controllers <interface\_id>**
- You must config the clk rate on the DCE side
  - R1(config-if)# **clock rate <rate in bps>**

## Point-to-point



```
R2#show ip ospf neighbor

Neighbor ID      Pri   State          Dead Time    Address      Interface
1.1.1.1          0     FULL/ -        00:00:31    192.168.1.1  Serial2/0
3.3.3.3          1     2WAY/DROTHER  00:00:39    192.168.2.2  GigabitEthernet0/0
4.4.4.4          1     FULL/DR       00:00:38    192.168.2.3  GigabitEthernet0/0
5.5.5.5          1     FULL/BDR      00:00:31    192.168.2.4  GigabitEthernet0/0
```

- Not considered any router type since it is a point-to-point connection

## Config Network Type

```
R1(config-if)#ip ospf network ?
broadcast          Specify OSPF broadcast multi-access network
non-broadcast      Specify OSPF NBMA network
point-to-multipoint Specify OSPF point-to-multipoint network
point-to-point     Specify OSPF point-to-point network
```

- Can configure network type on an interface
  - R1(config-if)# **ip ospf network <network-type>**

- For example, if 2 routers directly connected with an Ethernet link, no need DR/BDR. Can config a point-to-point network type instead
- Note: Not all network types work on all link types
  - E.g. Serial link cannot use the broadcast network type

Broadcast	Point-to-point
Default on <b>Ethernet, FDDI</b> interfaces	Default on <b>HDLC, PPP</b> (serial) interfaces
DR/DBR elected	No DR/BDR
Neighbors dynamically discovered	Neighbors dynamically discovered
Default timers: Hello 10, Dead 40	Default timers: Hello 10, Dead 40

(**Non-broadcast** network type default timers = Hello 30, Dead 120)

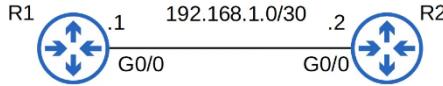
## OSPF Neighbour Requirements

1. Area number must match
2. Interfaces must be in the same subnet
3. OSPF process must not be shutdown
4. OSPF router IDs must be unique
5. Hello and Dead timers must match
6. Authentication settings must match

The following rules still can become OSPF neighbours, but OSPF doesn't operate properly

7. IP MTU must match
8. OSPF network type must match

Point 3 - OSPF process must not be shut down



```
R2(config)#router ospf 1
R2(config-router)#shutdown
R2(config-router)#
*Aug 23 03:43:31.719: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: Int
R2(config-router)#do show ip ospf neighbor
R2(config-router)#
R2(config-router)#no shutdown
R2(config-router)#
*Aug 23 03:49:52.931: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on GigabitEthernet0/0 from LOADING to FULL, Loading Done
R2(config-router)#do show ip ospf neighbor
Neighbor ID      Pri  State          Dead Time    Address      Interface
192.168.1.1      1    FULL/DR       00:00:38    192.168.1.1   GigabitEthernet0/0
R2(config-router)#

```

- Can shutdown OSPF without deleting the configurations

#### Point 4 - Router IDs must be unique

```
R2(config-router)#router-id 192.168.1.1
% OSPF: Reload or use "clear ip ospf process" command, for this to take effect
R2(config-router)#end
R2#clear ip
*Aug 23 03:57:58.835: %SYS-5-CONFIG_I: Configured from console by console
R2#clear ip ospf process
Reset ALL OSPF processes? [no]: yes
R2#
*Aug 23 03:58:04.055: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: Interface down or d
R2#
*Aug 23 03:58:06.495: %OSPF-4-DUP_RTRID_NBR: OSPF detected duplicate router-id 192.168.1.1 from 192.168.1.1 on interface GigabitEthernet0/0
R2#show ip ospf neighbor
R2#
R2(config-router)#no router-id
R2(config-router)#
*Aug 23 04:10:20.207: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on GigabitEthernet0/0 from LOADING to FULL, Loading Done
R2(config-router)#do show ip ospf neighbor
Neighbor ID      Pri  State          Dead Time    Address      Interface
192.168.1.1      1    FULL/DR       00:00:35    192.168.1.1   GigabitEthernet0/0
R2(config-router)#

```

- Why don't need to reset?
  - R2 does not have any other neighbours, so don't need to worry about affecting other routers

#### Point 5 - Hello and Dead timers must match

```
R2(config-if)#ip ospf hello-interval ?
<1-65535>  Seconds
R2(config-if)#ip ospf hello-interval 5
R2(config-if)#ip ospf dead-interval ?
<1-65535>  Seconds
minimal      Set to 1 second
R2(config-if)#ip ospf dead-interval 20
R2(config-if)#
*Aug 23 04:29:30.623: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: Dead timer expired
R2(config-if)#do show ip ospf neighbor
R2(config-if)#
R2(config-if)#no ip ospf hello-interval
R2(config-if)#no ip ospf dead-interval
R2(config-if)#
*Aug 23 04:31:32.727: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on GigabitEthernet0/0 from LOADING to FULL, Loading Done
R2(config-if)#do show ip ospf neighbor
Neighbor ID      Pri  State          Dead Time    Address      Interface
192.168.1.1      1    FULL/BDR     00:00:35    192.168.1.1   GigabitEthernet0/0
R2(config-if)#

```

- Can manually config the Hello and Dead timers

- Can reset them to default values by using "no"

#### Point 6 - Authentication settings must match

```
R2(config-if)#ip ospf authentication-key jeremy
R2(config-if)#ip ospf authentication
R2(config-if)#
*Aug 23 04:56:28.435: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: D
R2(config-if)#do show ip ospf neighbor
R2(config-if)#

R2(config-if)#no ip ospf authentication
R2(config-if)#no ip ospf authentication-key jeremy
*Aug 23 04:59:37.315: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on GigabitEthernet0/0 from LOADING to FULL, Loading Done
R2(config-if)#do show ip ospf neighbor

Neighbor ID      Pri  State          Dead Time    Address      Interface
192.168.1.1      1    FULL/BDR     00:00:34    192.168.1.1   GigabitEthernet0/0
R2(config-if)#

```

- Passwords on both routers must match

#### Point 7 - IP MTU must match

```
R2(config-if)#ip mtu ?
<68-1500> MTU (bytes)

R2(config-if)#ip mtu 1400
R2(config-if)#do show ip ospf neighbor

Neighbor ID      Pri  State          Dead Time    Address      Interface
192.168.1.1      1    FULL/BDR     00:00:34    192.168.1.1   GigabitEthernet0/0
R2(config-if)#do clear ip ospf process
Reset ALL OSPF processes? [no]: yes
R2(config-if)#
*Aug 23 05:16:07.474: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on GigabitEthernet0/0 from FULL to DOWN, Neighbor Down: Interface down or detached
R2(config-if)#do show ip ospf neighbor

Neighbor ID      Pri  State          Dead Time    Address      Interface
192.168.1.1      1    EXSTART/DR   00:00:38    192.168.1.1   GigabitEthernet0/0
R2(config-if)#
*Aug 23 05:21:12.946: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on GigabitEthernet0/0 from EXSTART to DOWN, Neighbor Down: Too many retransmissions
R2(config-if)#
*Aug 23 05:22:12.946: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on GigabitEthernet0/0 from DOWN to DOWN, Neighbor Down: Ignore timer expired
R2(config-if)#no ip mtu
R2(config-if)#
*Aug 23 05:25:49.362: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.1 on GigabitEthernet0/0 from LOADING to FULL, Loading Done

```

- Default MTU value = 1500

#### Point 8 - OSPF network type must match

```

R1#show ip ospf neighbor

Neighbor ID      Pri  State            Dead Time     Address          Interface
192.168.1.2        1    FULL/BDR       00:00:31   192.168.1.2    GigabitEthernet0/0

R1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
      + - replicated route, % - next hop override

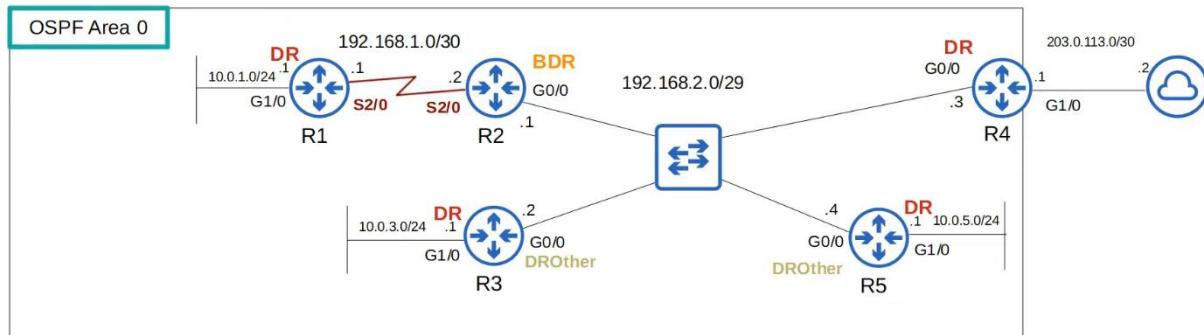
Gateway of last resort is not set

      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/30 is directly connected, GigabitEthernet0/0
L        192.168.1.1/32 is directly connected, GigabitEthernet0/0
R1#

```

- R2 loopback address is missing from the routing table
- A bit difficult to spot

## OSPF LSA Types



- The OSPF LSDB is made up of LSAs
- There are 11 types of LSA, but only 3 needed for CCNA
  - Type 1 (Router LSA)
  - Type 2 (Network LSA)
  - Type 5 (AS External LSA)
- Type 1 (Router LSA)

- Every OSPF router generates this type of LSA
- It identifies the router using its router ID
- It also lists networks attached to the router's OSPF-activated interfaces
- Type 2 (Network LSA)
  - Generated by the DR of each 'multi-access' network (e.g the broadcast network type)
  - List the routers which are attached to the multi-access network
- Type 5 (AS-External LSA)
  - Generated by ASBRs to describe routes to destinations outside of the AS (OSPF domain)

```
R1#show ip ospf database

OSPF Router with ID (1.1.1.1) (Process ID 1)

Router Link States (Area 0)

Link ID      ADV Router      Age      Seq#      Checksum Link count
1.1.1.1      1.1.1.1        1396     0x80000002 0x00FE8D 4
2.2.2.2      2.2.2.2        932      0x80000005 0x00753F 4
3.3.3.3      3.3.3.3        974      0x80000004 0x00AD70 2
4.4.4.4      4.4.4.4        975      0x80000005 0x004CC2 2
5.5.5.5      5.5.5.5        976      0x80000004 0x00D212 3

Net Link States (Area 0)

Link ID      ADV Router      Age      Seq#      Checksum
192.168.2.3  4.4.4.4        932      0x80000002 0x00740D

Type-5 AS External Link States

Link ID      ADV Router      Age      Seq#      Checksum Tag
0.0.0.0      4.4.4.4        273      0x80000002 0x00C0E0 1

R1#
```

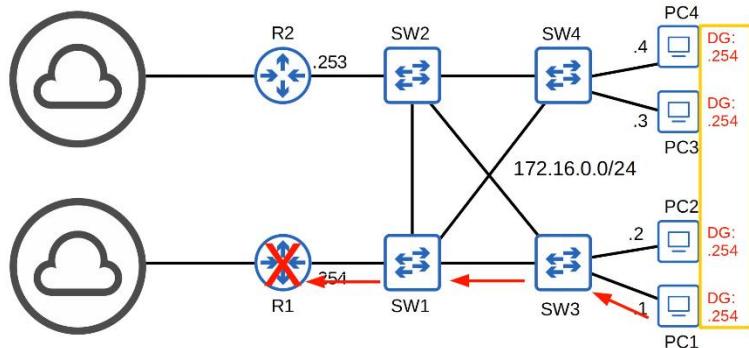
## FHRP

### Things covered

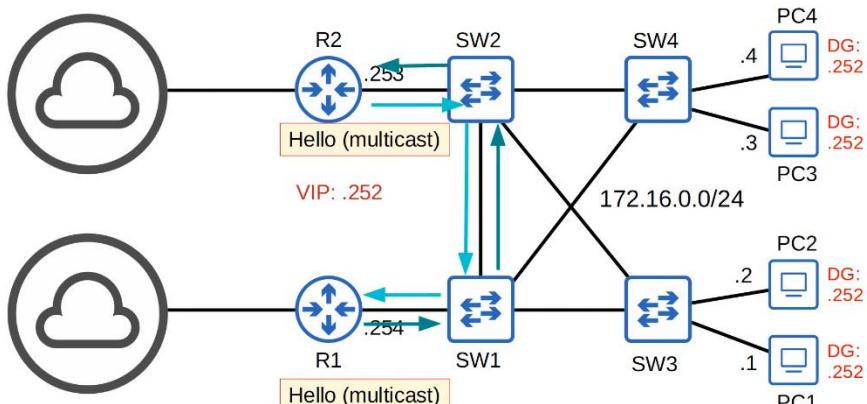
- The purpose of FHRPs
- HSRP (Hot Standby Router Protocol)

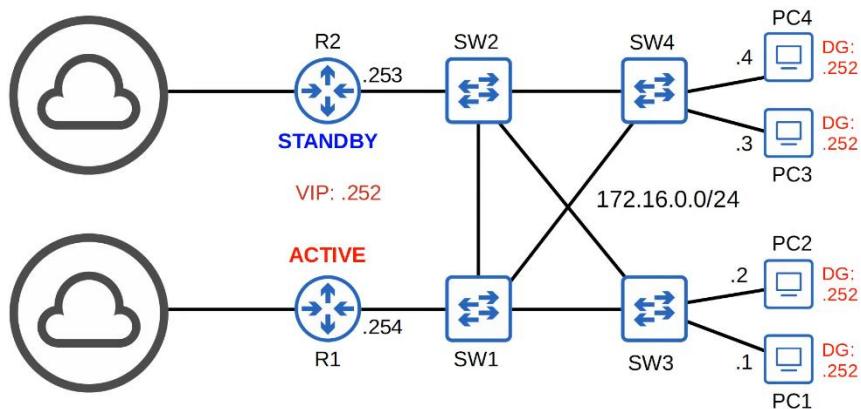
- VRRP (Virtual Router Redundancy Protocol)
- GLBP (Gateway Load Balancing Protocol)
- HSRP config

### Purpose of FHRP

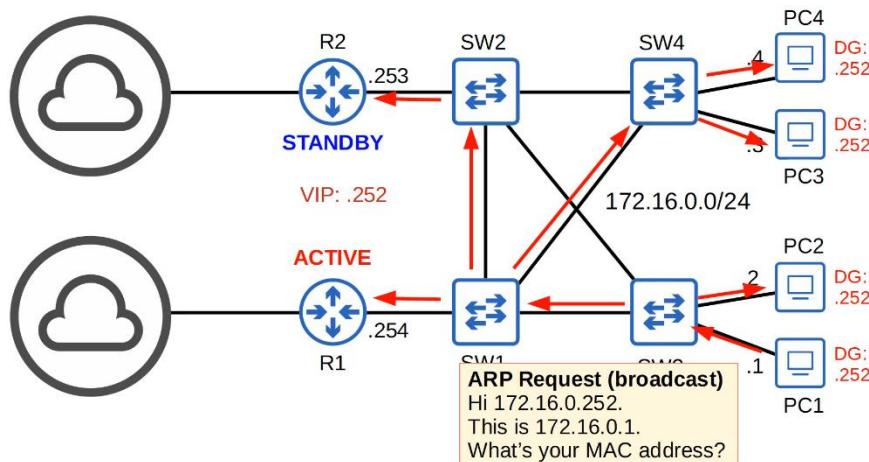


- Problem: Although there is an alternate route to the internet, default gateway of the PCs is to R1
- FHRP is designed to protect the default gateway used on a subnetwork by allowing 2 or more routers to provide backup for that address; in the event of a failure of an active router, the backup router will take over the address, usually within a few seconds

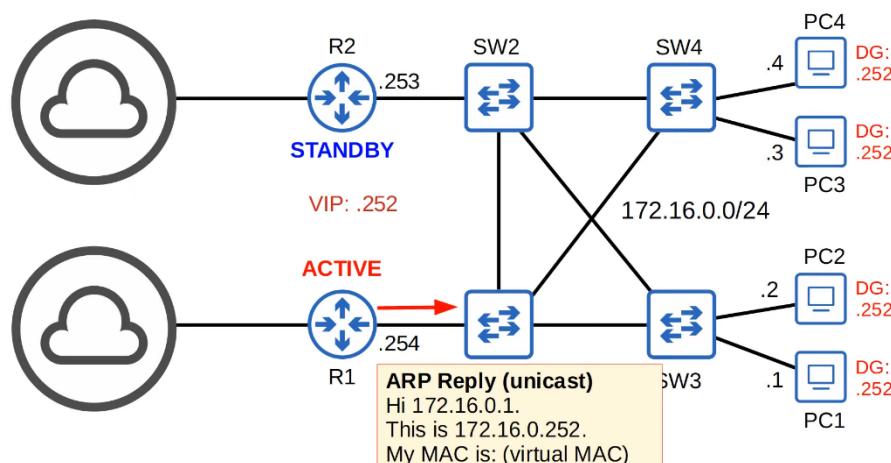


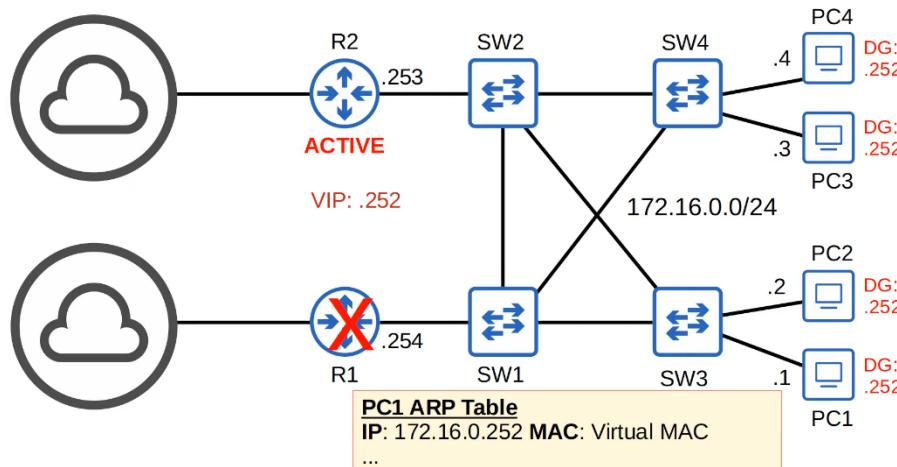
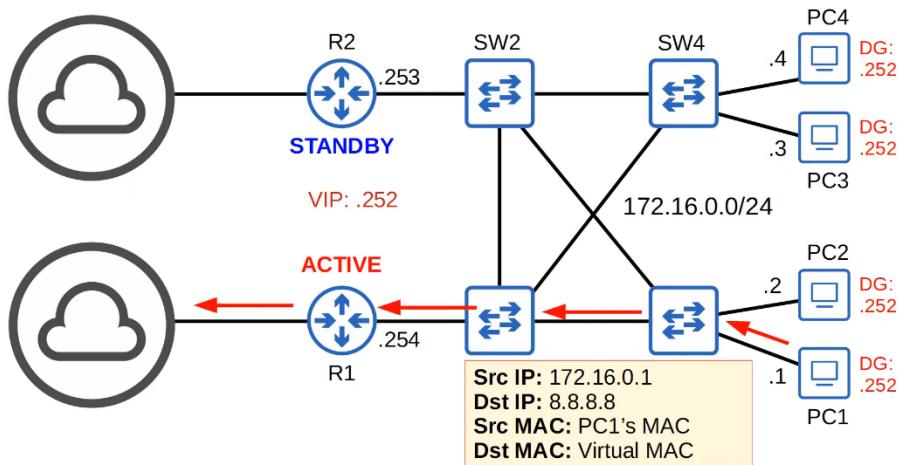


- The routers negotiate the roles by sending multicast messages to each other

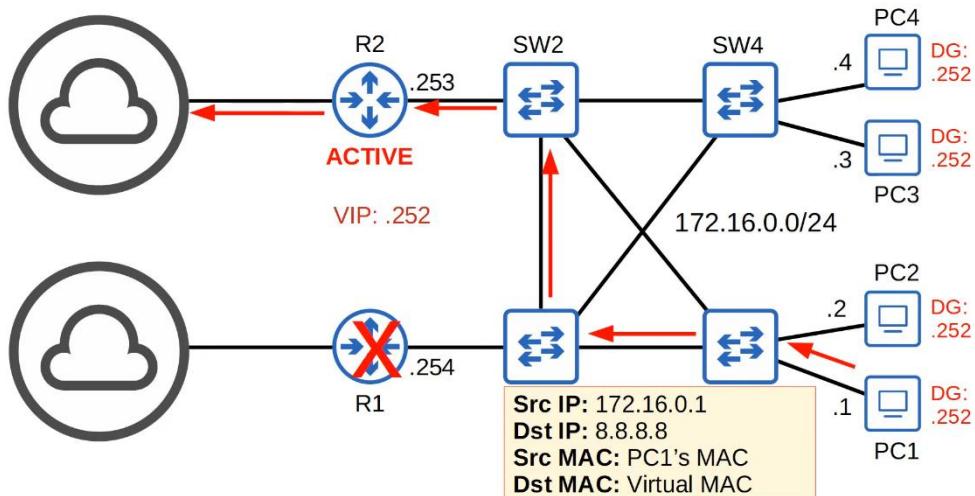
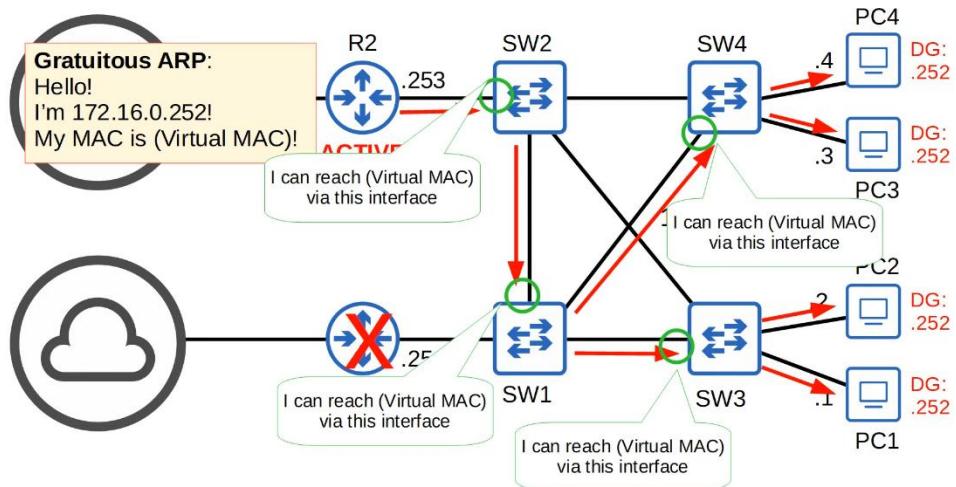


- PC1 wants to send traffic to a destination in another network, needs to send ARP request to get the MAC address of the default gateway (.252)





- When R1 fails, no need to change anything for the PC
  - Their ARP table maps the virtual IP to the virtual MAC
- R2 will realise that it is not receiving Hello messages from R1 and assumes that it is down, R2 will then become the active router
- Need to change the switches MAC address table
- R2 will send a gratuitous ARP
  - ARP replies that are sent without receiving an ARP request
  - Frames are broadcast to FFFF.FFFF.FFFF (instead of unicast)



- If R1 comes back online, it will become standby
- FHRPs are pre-emptive
  - Current active routers will not auto give up its role, even if the former active router returns
- However, can change this setting

## Overview

- A virtual IP is configured on the two routers, and a virtual MAC is generated for the virtual IP (each FHRP uses a different format for the virtual MAC)

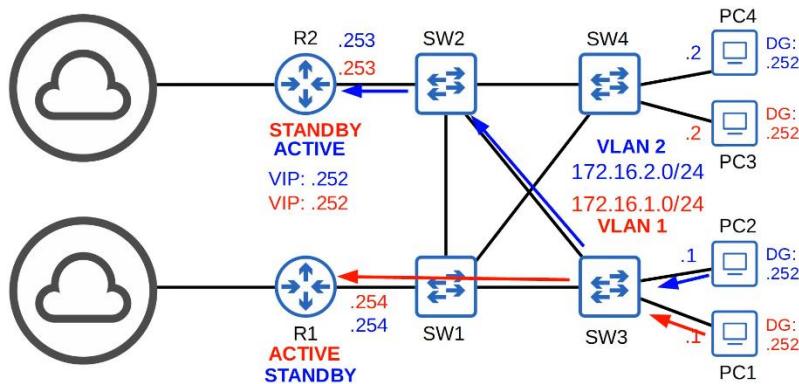
- An active router and a standby router are elected. (different FHRPs use different terms)
- End hosts in the network are configured to use the virtual IP as their default gateway.
- The active router replies to ARP requests using the virtual MAC address, so traffic destined for other networks will be sent to it.
- If the active router fails, the standby becomes the next active router.

The new active router will send gratuitous ARP messages so that switches will update their MAC address tables. It now functions as the default gateway.

- If the old active router comes back online, by default it won't take back its role as the active router. It will become the standby router.
- You can configure 'preemption', so that the old active router does take back its old role.

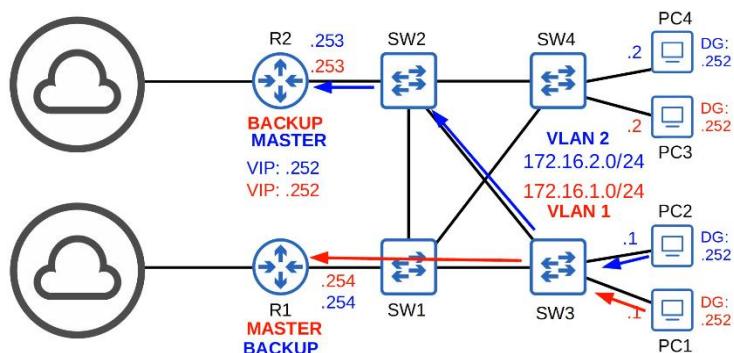
### **HSRP (Hot Standby Router Protocol)**

- Cisco proprietary
- Standby and Active router are selected
- 2 versions
  - Version 1
  - Version 2: support IPv6 and more groups can be configured
- Multicast IPv4
  - V1: 224.0.0.2
  - V2: 224.0.0.102
- Virtual MAC address
  - V1: 0000.0c07.acxx
  - V2: 0000.0c9f.fxxx
  - 'x': HSRP group number
- In a situation with multiple subnets/VLANs, can configure a different active router in each subnet/VLAN to load balance



## VRRP (Virtual Router Redundancy Protocol)

- Open standard
- A 'master' and 'backup' router are elected
- Multicast IPv4: 224.0.0.18
- Virtual MAC address: 0000.5e00.01xx (xx: VRRP group number)
- In a situation with multiple subnets/VLANs, can configure a different master router in each subnet/VLAN to load balance



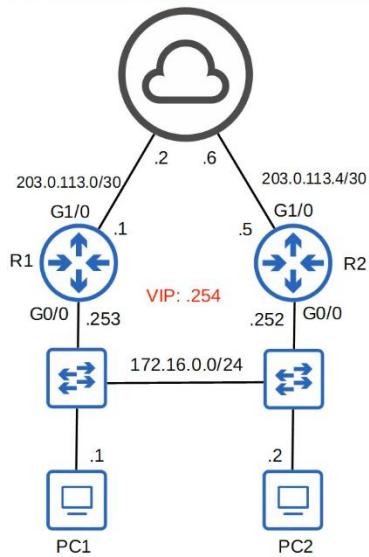
- Note about VLAN and Subnet
  - All hosts in VLAN1 is in the same subnet (172.16.1.0/24)
  - All hosts in VLAN2 is in the same subnet (172.16.2.0/24)
  - Subnets divide the network in layer 3 and VLANs divide the network in layer 2
  - They work together, with each subnet being its own VLAN

## GLBP (Gateway Load Balancing Protocol)

- Cisco proprietary
- Load balance across multiple routers within a single subnet
- An AVG (Active Virtual Gateway) is elected
- Up to 4 AVFs (Active Virtual Forwarders) are assigned by the AVG (the AVG itself can be an AVF, too)
- Each AVF acts as the default gateway for a portion of the hosts in the subnet
- Multicast IPv4: 224.0.0.102
- Virtual MAC address: 0007.b400.**XXYY**
  - **X**: GLBP group number
  - **Y**: AVF number

FHRP	Terminology	Multicast IP	Virtual MAC	Cisco proprietary?
HSRP	Active/Standby	v1: 224.0.0.2 v2: 224.0.0.102	v1: 0000.0c07.acXX v2: 0000.0c9f.fXXX	Yes
VRRP	Master/Backup	224.0.0.18	0000.5e00.01XX	No
GLBP	AVG / AVF	224.0.0.102	0007.b400.XXYY	Yes

## Config



```
R1(config)#interface g0/0
R1(config-if)#standby ?
<0-255>      group number
authentication Authentication
bfd           Enable HSRP BFD
[...]
```

```
R1(config-if)#standby version 2
R1(config-if)#standby ?
<0-4095>      group number
authentication Authentication
bfd           Enable HSRP BFD
```

```
R1(config-if)#standby 1 ?
authentication Authentication
follow        Name of HSRP group to follow
ip            Enable HSRP IPv4 and set the virtual IP address
ipv6          Enable HSRP IPv6
mac-address   Virtual MAC address
name          Redundancy name string
preempt       Overthrow lower priority Active routers
priority      Priority level
timers        Hello and hold timers
track         Priority tracking
```

- Group number must match on the routers

```
R1(config-if)#standby 1 ip 172.16.0.254
R1(config-if)#
R1(config-if)#standby 1 priority ?
<0-255>  Priority value
R1(config-if)#standby 1 priority 200
R1(config-if)#
R1(config-if)#standby 1 preempt
R1(config-if)#[...]
```

The **active router** is determined in this order:  
 1 – Highest priority (default 100)  
 2 – Highest IP address

**Preempt** causes the router to take the role of active router, even if another router already has the role.

Only necessary on the router you want to become active

```
R2(config-if)#standby version 2
R2(config-if)#
R2(config-if)#standby 1 ip 172.16.0.254
R2(config-if)#
R2(config-if)#standby 1 priority 50
R2(config-if)#
R2(config-if)#standby 1 preempt
```

HSRP version 1 and version 2 are not compatible.  
 If R1 uses version 2, R2 must use version 2 also.

- 'priority 50'
  - Not needed since set the priority on R1 to higher than default
- 'preempt'

- Not needed since only for the active router, can do to standardise the config btw both routers

```
R1#show standby
GigabitEthernet0/0 - Group 1 (version 2)
  State is Active
    2 state changes, last state change 00:16:30
  Virtual IP address is 172.16.0.254
  Active virtual MAC address is 0000.0c9f.f001
    Local virtual MAC address is 0000.0c9f.f001 (v2 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 1.536 secs
  Preemption enabled
  Active router is local
  Standby router is 172.16.0.252, priority 50 (expires in 9.280 sec)
  Priority 200 (configured 200)
  Group name is "hsrp-Gi0/0-1" (default)
R1#
```

```
R2#show standby
GigabitEthernet0/0 - Group 1 (version 2)
  State is Standby
    1 state change, last state change 00:17:05
  Virtual IP address is 172.16.0.254
  Active virtual MAC address is 0000.0c9f.f001
    Local virtual MAC address is 0000.0c9f.f001 (v2 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 1.472 secs
  Preemption enabled
  Active router is 172.16.0.253, priority 200 (expires in 10.160 sec)
    MAC address is 0c9f.6041.8800
  Standby router is local
  Priority 50 (configured 50)
  Group name is "hsrp-Gi0/0-1" (default)
R2#
```

- R1(config-if)# **standby version 2**
- R1(config-if)# **standby <group-num> ip <virtual-ip>**
- R1(config-if)# **standby <group-num> priority <priority>**
- R1(config-if)# **standby <group-num> preempt**

# TCP/UDP

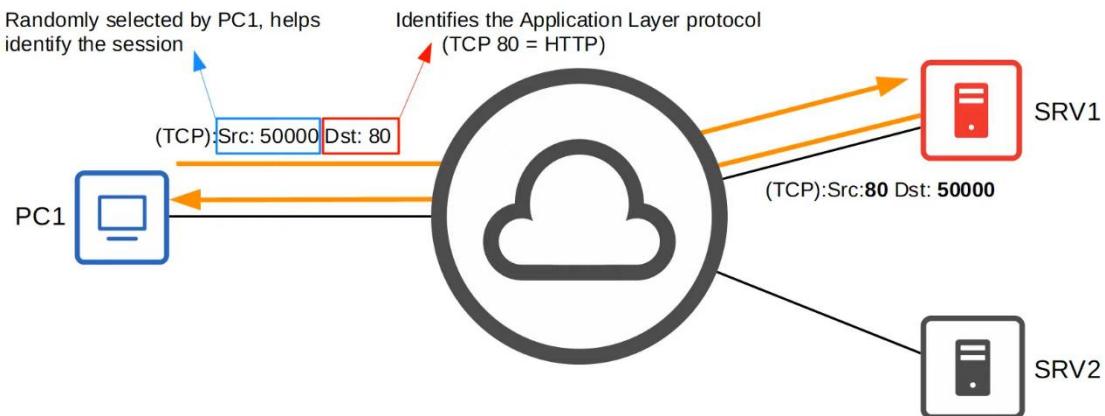
Things covered

- Basics of Layer 4
- TCP (Transmission Control Protocol)
- UDP (User Datagram Protocol)
- Comparing TCP and UDP

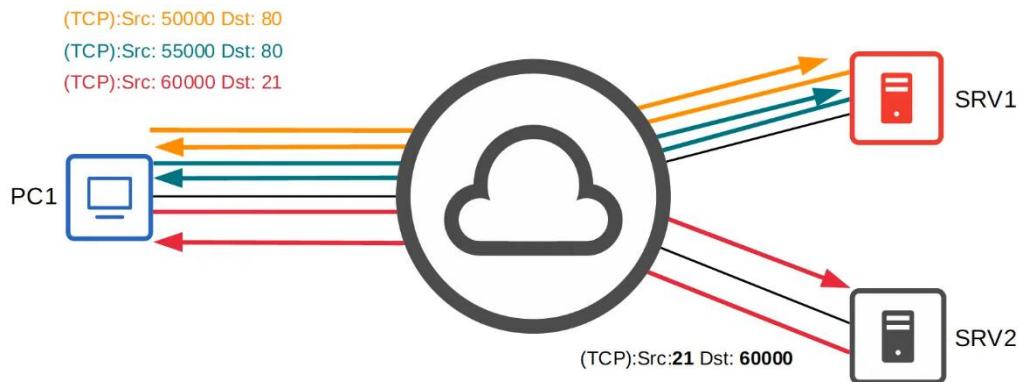
## **Functions of Layer 4 (Transport Layer)**

- Provides transparent transfer of data btw end hosts
- Provides (or don't provide) various services to applications
  - Reliable data transfer
  - Error recovery
  - Data sequencing
  - Flow control
- Provide Layer 4 addressing (port numbers)
  - Port: NOT the physical interfaces/ports on network devices
  - Identify the Application layer protocol
  - Provides session multiplexing
  - The following ranges have been designated by IANA (Internet Assigned Numbers Authority)
    - **Well known** port numbers: 0 - 1023
    - **Registered** port numbers: 1024 - 49151
    - **Ephemeral/private/dynamic** port numbers: 49152 - 65535

## **Port Numbers / Session Multiplexing**



- A session is an exchange of data between 2 or more communicating devices
- The response from SRV1 to PC1 will be opposite for the Src and Dst ports
  - Allows PC1 to know it is from that particular session



- Can have multiple sessions running at the same time

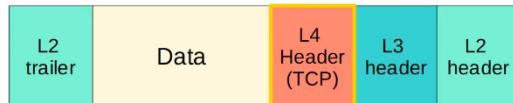
## TCP

- It is connection oriented
  - The 2 hosts communicate to establish a connection before sending data
  - Once established, data exchange begins
- Provides reliable connection
  - Destination host must acknowledge that it received each TCP segment (layer 4 PDU)
  - If not acknowledged, it is sent again

- Provides sequencing
  - Sequence numbers in the TCP header allow destination hosts to put segments in the correct order even if they arrive out of order
- Provides flow control
  - Destination host can tell source host to increase/decrease the rate that data is sent

## TCP Header

TCP segment header																									
Offsets	Octet	0				1				2				3											
Octet	Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	0	Source port												Destination port											
4	32	Sequence number																							
8	64	Acknowledgment number (if ACK set)																							
12	96	Data offset	Reserved 000	N S	C W R	E C E	U R G	A C K	P C H	R S T	S Y N	F I N	Window Size												
16	128	Checksum												Urgent pointer (if URG set)											
20	160	Options (if data offset > 5. Padded at the end with "0" bytes if necessary.)												...											
...	...													...											

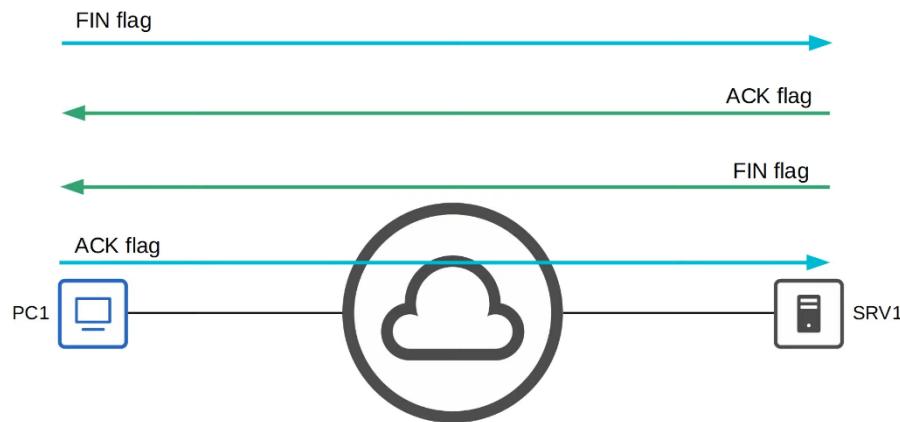


- Source/Destination Port
  - 16 bits = 65536 available port numbers
- Sequence/Ack Number
  - Provide sequencing and reliable communication
- Flags: ACK, SYN, FIN
  - Used to establish and terminate communications
- Window size
  - Used for flow control

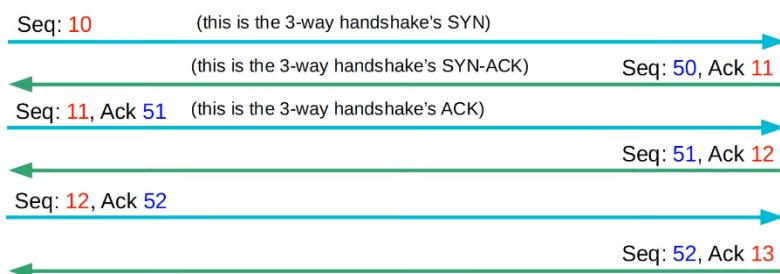
## Establishing Connections: 3 way handshake



### Terminating Connections: 4 way handshake



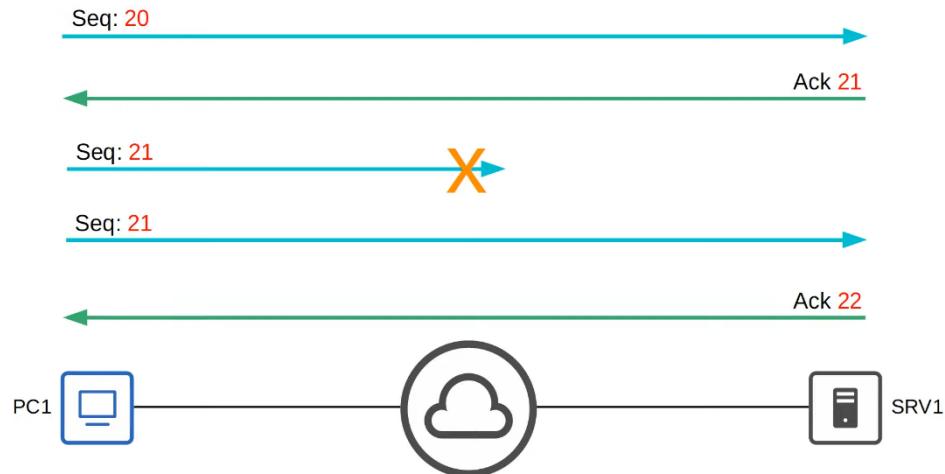
### Sequencing/Acknowledgment



- Hosts set a random initial sequence number

- Forward acknowledgment is used to indicate the sequence number of the next segment it expects to receive

## Retransmission



## Flow Control: Window size



- Acknowledging every single segment is inefficient
- The TCP header's 'Window Size' field allows more data to be sent before an acknowledgment is required
- A 'sliding window' can be used to dynamically adjust how large the window size is
- Note:
  - In all these examples, simple sequence numbers are used. In reality, the seq numbers can get much larger and do not increase by 1 with each message.

- For CCNA, just understand the concepts and don't worry about the exact numbers

## UDP

- Not connection oriented
  - Sending host does not establish a connection with destination host before sending data
  - Data is simply sent
- Does not provide reliable communication
  - Ack are not sent for received segments
  - If segment lost, no mechanism for retransmission
  - Segments are sent 'best-effort'
- Does not provide sequencing
  - No seq number field in header
  - If arrive out of order, no mechanism to put them back in order
- Does not provide flow control
  - No mechanism like TCP's window size to control flow of data

UDP datagram header																																	
Offsets Octet		0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port								Destination port																							
4	32	Length								Checksum																							

## Comparing TCP & UDP

- TCP provides more features than UDP, but at the cost of additional overhead.
- For applications that require reliable communications (for example downloading a file), TCP is preferred.
- For applications like real-time voice and video, UDP is preferred.
- There are some applications that use UDP, but provide reliability etc within the application itself.
- Some applications use both TCP & UDP, depending on the situation.

TCP	UDP
Connection-oriented	Connectionless
Reliable	Unreliable
Sequencing	No sequencing
Flow control	No flow control
Use for downloads, file sharing, etc	Used for VoIP, live video, etc

## Port Numbers

### TCP

- FTP Data: 20
- FTP Control: 21
- SSH: 22
- Telnet: 23
- SMTP: 25
- HTTP: 80
- POP3: 110
- HTTPS: 443

### UDP

- DHCP server: 67
- DHCP client: 68
- TFTP: 69
- SNMP agent: 161
- SNMP manager: 162

- Syslog: 514

## UDP & TCP

- DNS: 53

# IPv6 Addressing

## IPv6 (Part 1)

### Things covered

- Hexadecimal review
- Why IPv6
- Basics of IPv6
- Config

### What about IPv5

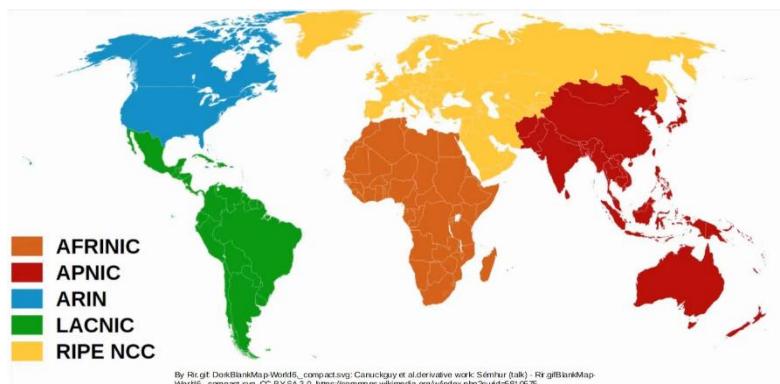
- 'Internet Stream Protocol', never introduced for public use
- Used a value of 5 in Version field of IP header
- So, when successor of IPv4 was developed, named IPv6

### Hexadecimal

- Binary / Base 2 / 0b01
- Decimal / Base 10 / 0d23
- Hexadecimal / Base 16 / 0xA3

## Why IPv6

- Not enough IPv4 addresses available
- 4,294,967,296 ( $2^{32}$ ) available IPv4 addresses
- VLSM, private IPv4 addresses, and NAT has been used to conserve the use of IPv4 address space
- Those are short term solutions
- Long term solution is IPv6
- IPv4 address assignments are controlled by IANA (Internet Assigned Numbers Authority)
- IANA distributes IPv4 address space to various RIRs (Regional Internet Registries), which then assign them to companies that need them



## IPv6

- Address is 128 bits
- Example
  - 2001:0D88:5917:EABD:6562:C923:59BD / 64

## Shortening IPv6 addresses

- **Leading 0s** can be removed

2001:~~0~~D88:~~000~~A:~~00~~1B:~~20~~A1:~~00~~20:~~00~~80:34BD



2001:DB8:A:1B:20A1:20:80:34BD

- **Consecutive quartets of all 0s** can be replaced by '::'

2001:0DB8:**0000:0000:0000:0000**:0080:34BD



2001:0DB8::0080:34BD



Combine both methods

2001:DB8::80:34BD

- However, can only do it once
- Else don't know how many quartets of 0 for each '::'

2001:0000:0000:0000:20A1:0000:0000:34BD

How many quartets of 0 are here?

How many quartets of 0 are here?

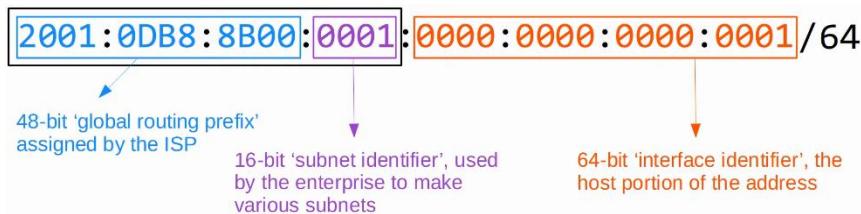
2001:**0000:0000:0000**:20A1:0000:0000:34BD

### Expanding shortened IPv6 address

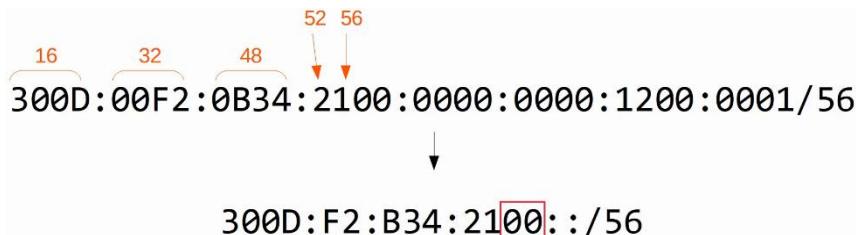
- Put leading 0s where needed
- If double colon used, replace with all 0s quartet
- Make sure there is 8 quartets

### Finding the IPv6 prefix (global unicast addresses)

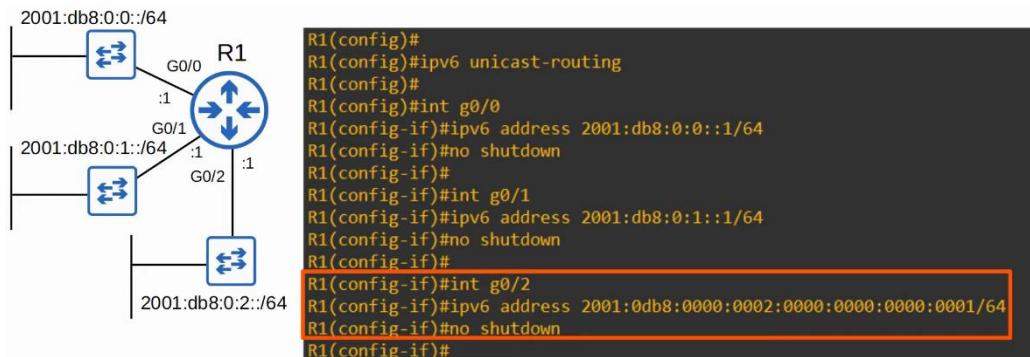
- Typically, an enterprise requesting IPv6 addresses from their ISP will receive a /48 block
- Typically, IPv6 subnets use a /64 prefix length
- That means an enterprise has 16 bits to use to make subnets
- The remaining 64 bits can be used for hosts



## Examples



## Config



```
R1#show ipv6 interface brief
GigabitEthernet0/0      [up/up]
    FE80::EF8:22FF:FE36:8500
    2001:DB8::1
GigabitEthernet0/1      [up/up]
    FE80::EF8:22FF:FE36:8501
    2001:DB8:0:1::1
GigabitEthernet0/2      [up/up]
    FE80::EF8:22FF:FE36:8502
    2001:DB8:0:2::1
GigabitEthernet0/3      [administratively down/down]
    unassigned
R1#
```

- Link-local Addresses
  - Auto enabled on IPv6 interfaces

#### Commands

- R1(config)# ipv6 unicast-routing
- R1(config-if)# ipv6 <ipv6 address>/<subnet mask>

## IPv6 (Part 2)

#### Things covered

- IPv6 Config
  - Modified EUI-64
- IPv6 address types
  - Global unicast
  - Unique local
  - Link local
  - Multicast
  - Others

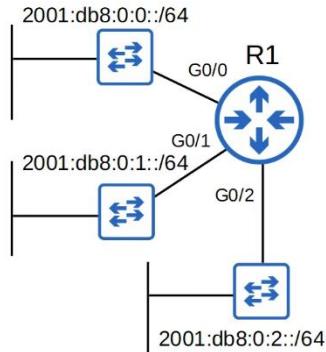
## Configuring IPv6 addresses (EUI-64)

- EUI: Extended Unique Identifier
- (Modified) EUI-64 is a method of converting a MAC address (48 bits) into a 64-bit interface identifier
- This interface identifier can then become the 'host portion' of a /64 IPv6 address
- How to convert MAC address
  1. Divide the MAC address in half
  2. Insert FFFE in the middle
  3. Invert the 7th MSB

**1234 5678 90AB → 1234 56 | 78 90AB**

**1234 56FF FE78 90AB**

**1234 56FF FE78 90AB → 1034 56FF FE78 90AB**



```
R1(config)#int g0/0
R1(config-if)#ipv6 address 2001:db8::/64 eui-64
R1(config-if)#no shutdown
R1(config-if)#
R1(config-if)#int g0/1
R1(config-if)#ipv6 address 2001:db8:0:1::/64 eui-64
R1(config-if)#no shutdown
R1(config-if)#
R1(config-if)#int g0/2
R1(config-if)#ipv6 address 2001:db8:0:2::/64 eui-64
R1(config-if)#no shutdown
```

```

R1#show interfaces g0/0
GigabitEthernet0/0 is administratively down, line protocol is down
  Hardware is iGbE, address is 0cf8.2236.8500 (bia 0cf8.2236.8500)

R1#show interfaces g0/1
GigabitEthernet0/1 is administratively down, line protocol is down
  Hardware is iGbE, address is 0cf8.2236.8501 (bia 0cf8.2236.8501)

R1#show interfaces g0/2
GigabitEthernet0/2 is administratively down, line protocol is down
  Hardware is iGbE, address is 0cf8.2236.8502 (bia 0cf8.2236.8502)

R1(config-if)#do show ipv6 interface brief
GigabitEthernet0/0      [up/up]
  FE80::EF8:22FF:FE36:8500
  2001:DB8::EF8:22FF:FE36:8500
GigabitEthernet0/1      [up/up]
  FE80::EF8:22FF:FE36:8501
  2001:DB8:0:1:EF8:22FF:FE36:8501
GigabitEthernet0/2      [up/up]
  FE80::EF8:22FF:FE36:8502
  2001:DB8:0:2:EF8:22FF:FE36:8502
GigabitEthernet0/3      [administratively down/down]
  unassigned

```

- EUI-64 allows routers to auto generate an IPv6 address by expanding their MAC address to a 64 bit ID which is then combined with a specified IPv6 prefix

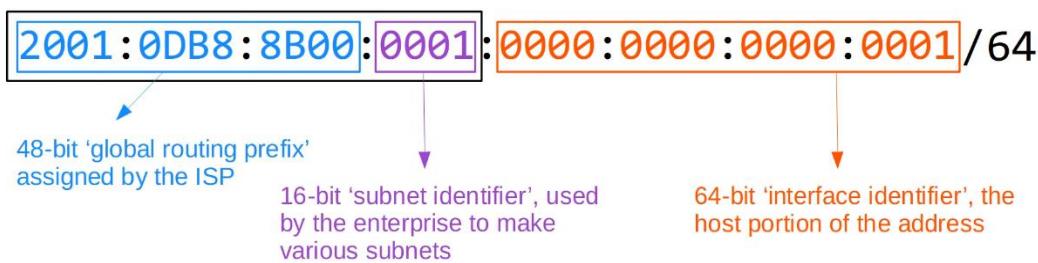
Why invert the 7th bit?

- MAC addresses can be divided into 2 types
  - UAA (Universally Administered Address)
    - Uniquely assigned to the device by the manufacturer
  - LAA (Locally Administered Address)
    - Manually assigned by an admin (with 'mac-address' command on the interface) or protocol
    - Doesn't have to be globally unique
- Can identify a UAA or LAA by the 7th bit of the MAC address
  - Called the U/L bit (Universal/Local bit)
  - 0: UAA
  - 1: LAA
- In the context of IPv6 addresses/EUI-64, the meaning of the U/L bit is reversed

- 0: MAC address of the EUI-64 interface was an LAA
- 1: MAC address of the EUI-64 interface was an UAA

## Global Unicast Address

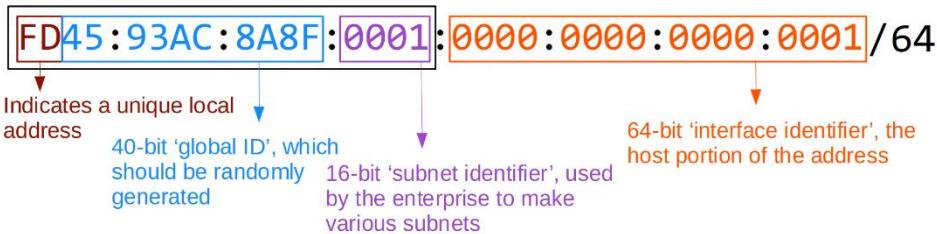
- Global unicast IPv6 addresses are public addresses which can be used over the Internet
- Must register to use them
  - Because they are public addresses, it is expected that they are globally unique
- Originally defined as the 2000::/3 block (2000:: to 3FFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)
- Now defined as all addresses which are not reserved for other purposes



## Unique Local Address

- Private addresses that cannot be used over the Internet
- Do not need to register them
  - Can be freely used within internal networks and don't need to be globally unique
  - Can't be routed over Internet, will be dropped by router
  - Note: Global ID should be unique so that addresses don't overlap when companies merge
- Uses the address block FC00::/7 (FC00:: to FDFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)

- However, a later update requires the 8th bit to be set to 1
  - So, the first 2 digits are FD

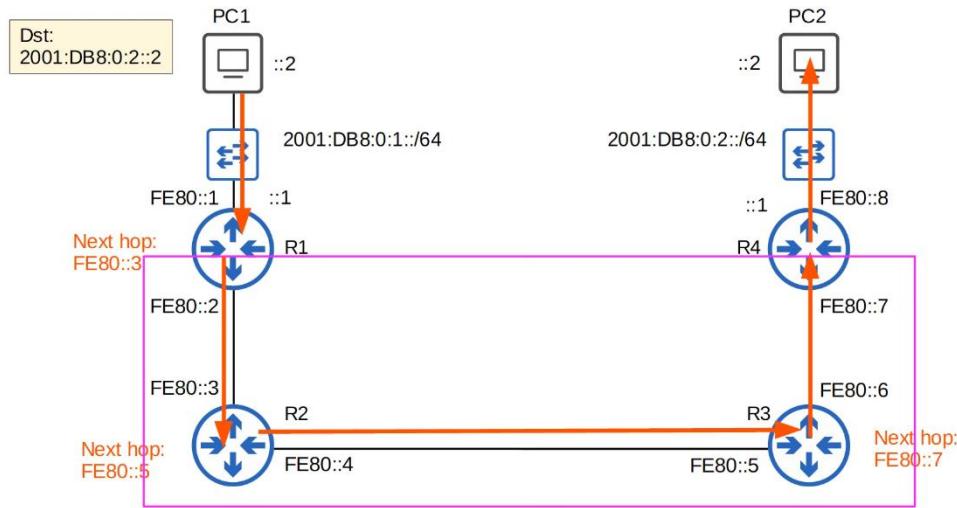


## Link Local Addresses

- Link-local IPv6 addresses are automatically generated on IPv6-enabled interfaces.
- Use command **R1(config-if)# ipv6 enable** on an interface to enable IPv6 on an interface.
  - Won't need to configure an actual IPv6 address on the interface
  - Will generate a Link Local IPv6 address
- Uses the address block FE80::/10 (FE80:: to FEBF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)
- However, the standard states that the 54 bits after FE80/10 should be all 0, so you won't see link local addresses beginning with FE9, FEA, or FEB. Only FE8.
- The interface ID is generated using EUI-64 rules.
- Link-local means that these addresses are used for communication within a single link (subnet).

Routers will not route packets with a link-local destination IPv6 address.

- Common uses of link-local addresses:
  - routing protocol peerings (OSPFv3 uses link-local addresses for neighbour adjacencies)
  - next-hop addresses for static routes
  - Neighbour Discovery Protocol (NDP, IPv6's replacement for ARP) uses link-local addresses to function



## Multicast Addresses

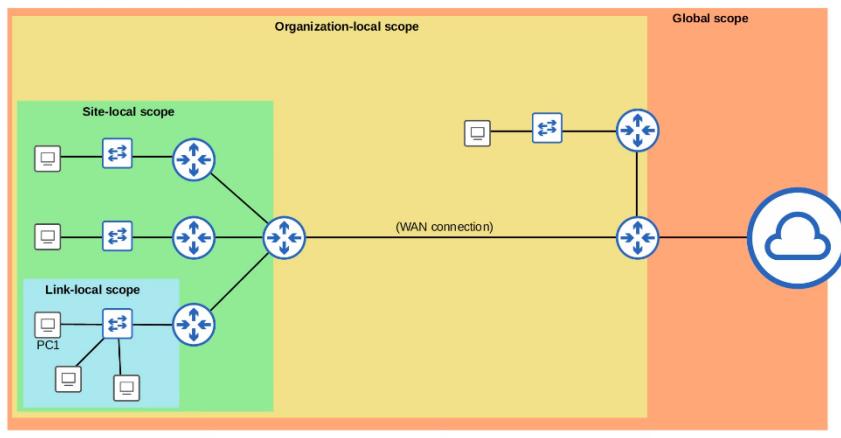
- Multicast addresses are one-to-many
  - 1 source to multiple destinations (that have joined the specific multicast group)
- IPv6 uses the range FF00::/8 for multicast
- IPv6 doesn't use broadcast (there is no 'broadcast address' in IPv6)

Purpose	IPv6 Address	IPv4 Address
All nodes/hosts (functions like broadcast)	FF02::1	224.0.0.1
All routers	FF02::2	224.0.0.2
All OSPF routers	FF02::5	224.0.0.5
All OSPF DRs/BDRs	FF02::6	224.0.0.6
All RIP routers	FF02::9	224.0.0.9
All EIGRP routers	FF02::A	224.0.0.10

## Multicast address scopes

- IPv6 defines multiple multicast 'scopes' which indicate how far the packet should be forwarded.
- The addresses in the previous slide all use the 'link-local' scope (FF02), which stays in the local subnet.

- IPv6 multicast scopes:
  - Interface-local (FF01): The packet doesn't leave the local device. Can be used to send traffic to a service within the local device.
  - Link-local (FF02): The packet remains in the local subnet. Routers will not route the packet between subnets.
  - Site-local (FF05): The packet can be forwarded by routers. Should be limited to a single physical location (not forwarded over a WAN)
  - Organization-local (FF08): Wider in scope than site-local (an entire company/organization).
  - Global (FF0E): No boundaries. Possible to be routed over the Internet.



```

R1#sh ipv6 int g0/0
GigabitEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::EF8:22FF:FE36:8500
    No Virtual link-local address(es):
      Global unicast address(es):
        2001:DB8::EF8:22FF:FE36:8500, subnet is 2001:DB8::/64 [EUI]
      Joined group address(es):
        FF02::1
        FF02::2
        FF02::1:FF36:8500
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 30000)
  ND advertised reachable time is 0 (unspecified)
  ND advertised retransmit interval is 0 (unspecified)
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  ND advertised default router preference is Medium
  Hosts use stateless autoconfig for addresses.

```

## Anycast Addresses

- New feature of IPv6
- Anycast is 'one to one-of-many'
- Multiple routers are configured with the same IPv6 address
  - They use a routing protocol to advertise the address
  - When hosts sends a packet to that destination address, routers will forward it to the nearest router configured with the IP address (based on routing metric)
- There is no specific address range for anycast addresses
  - Use a regular unicast address (global unicast/unique local etc) and specify it as an anycast address
  - R1(config-if)# **ipv6 address 2001:db8:1:1::99/128 anycast**

```

R1(config)#int g0/0
R1(config-if)#ipv6 address 2001:db8:1:1::99/128 anycast
R1(config-if)#
R1(config-if)#do show ipv6 interface g0/0
GigabitEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::EF8:22FF:FE36:8500
    No Virtual link-local address(es):
      Global unicast address(es):
        2001:DB8::EF8:22FF:FE36:8500, subnet is 2001:DB8::/64 [EUI]
        2001:DB8:1:1::99, subnet is 2001:DB8:1:1::99/128 [ANY]
    Joined group address(es):
      FF02::1
      FF02::2
      FF02::1:FF00:99
      FF02::1:FF36:8500
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 30000)
  ND advertised reachable time is 0 (unspecified)

```

- Although under 'Global unicast address(es)', it is an anycast address

## Other IPv6 addresses

- Unspecified IPv6 addresses
  - :: (all 0s)
  - Can be used when a device doesn't yet know its IPv6 address
  - IPv6 default routes are configured to ::/0
  - IPv4 equivalent: 0.0.0.0
- Loopback address
  - ::1
  - Used to test the protocol stack on the local device
  - Messages sent to this address are processed within the local device, but not sent to other devices
  - IPv4 equivalent: 127.0.0.0/8 address range

## Things covered

- Correction

- IPv6 Header
- Neighbour Discovery Protocol
- SLAAC
- IPv6 Routing

### Correction (IPv6 Address Representation)

- RFC 5952
  - Recommendation for IPv6 Address Text Representation
- Recommendations
  - Leading 0s must be removed
  - '::' must be used for the longest consecutive quartets of 0
    - If equal length, use the left one
  - Hexadecimal in lower-case

### Header

Offsets	Octet	0								1								2								3																
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31									
0	0	Version				Traffic Class								Flow Label																												
4	32	Payload Length																Next Header				Hop Limit																				
8	64	Source Address																Destination Address																								
12	96																																									
16	128																																									
20	160																																									
24	192	Source Address																	Destination Address																							
28	224																																									
32	256																																									
36	288																																									

- Fixed size: 40B
- Version
  - 4 bits
  - Indicates the version of IP used
  - Fixed value of 6 (0x0110) to indicate IPv6
- Traffic Class

- 8 bits
  - Used for QoS (Quality of Service), to indicate high-priority traffic
  - E.g. IP phone traffic, live video calls, etc, will have a Traffic Class value which gives them higher priority over other traffic
- Flow Label
  - 20 bits
  - Used to identify specific traffic 'flows' (communications btw a specific source and destination)
- Payload Length
  - 16 bits
  - Indicates the length of the payload (the encapsulated Layer 4 segment) in bytes
  - The length of the IPv6 header is not included since it is fixed at 40 bytes
- Next Header
  - 8 bits
  - Indicates the type of the 'next header' (header of the encapsulated segment), for example, TCP or UDP
  - Same function as the IPv4 header's 'Protocol' field
- Hop Limit
  - 8 bits
  - The value of the field is decremented by 1 by each router that forwards it
  - If value reaches 0, the packet is discarded
  - Same function as IPv4 header's 'TTL' field
- Source/Destination
  - 128 bits each

### Solicited-Node Multicast Address

- An IPv6 solicited-node multicast address is calculated from a unicast address
- Method to get the address



```

R1#sh ipv6 int g0/0
GigabitEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::EF8:22FF:FE36:8500
  No Virtual link-local address(es):
  Global unicast address(es):
    2001:DB8::EF8:22FF:FE36:8500, subnet is 2001:DB8::/64 [EUI]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF36:8500
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 30000)
  ND advertised reachable time is 0 (unspecified)
  ND advertised retransmit interval is 0 (unspecified)
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  ND advertised default router preference is Medium
  Hosts use stateless autoconfig for addresses.
  
```

## Neighbour Discovery Protocol

- NDP is a protocol used with IPv6
- Has various functions, 1 of those is to replace ARP, which is not used in IPv6
- The ARP-like function of NDP uses ICMPv6 and solicited-node multicast addresses to learn the MAC address of other hosts

*\*(ARP in IPv4 uses broadcast message)*

- 2 message types used

- Neighbour Solicitation (NS) - ICMPv6 type 135
- Neighbour Advertisement (NA) - ICMPv6 type 136

## Neighbour Solicitation



- Source IP: R1 G0/0 IP
- Destination IP: R2 solicited-node multicast address
- Source MAC: R1 G0/0 MAC
- Destination MAC: Multicast MAC based on R2's solicited-node address

```
> Frame 6: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface -, id 0
> Ethernet II, Src: ca:01:09:6d:00:08 (ca:01:09:6d:00:08), Dst: IPv6multicast_ff:78:9a:bc (33:33:ff:78:9a:bc)
> Internet Protocol Version 6, Src: 2001:db8::12:3456, Dst: ff02::1:ff78:9abc
> Internet Control Message Protocol v6
```

## Neighbour Advertisement



- Source IP: R2 G0/0 IP
- Destination IP: R1 G0/0 IP
- Source MAC: R2 G0/0 MAC
- Destination MAC: R1 G0/0 MAC

```
> Frame 7: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface -, id 0
> Ethernet II, Src: ca:02:09:7c:00:08 (ca:02:09:7c:00:08), Dst: ca:01:09:6d:00:08 (ca:01:09:6d:00:08)
> Internet Protocol Version 6, Src: 2001:db8::78:9abc, Dst: 2001:db8::12:3456
> Internet Control Message Protocol v6
```

- R2 knows the destination MAC address since it received the NS which has the source MAC address of R1 G0/0

## IPv6 Neighbour Table



```
R1#show ipv6 neighbor
```

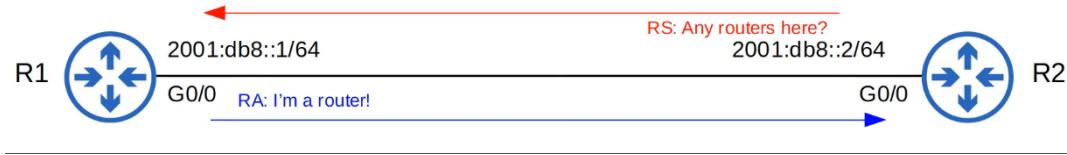
<b>IPv6 Address</b>	<b>Age</b>	<b>Link-layer Addr</b>	<b>State</b>	<b>Interface</b>
FE80::C802:9FF:FE7C:8	0	ca02.097c.0008	REACH	Gi0/0
2001:DB8::78:9ABC	0	ca02.097c.0008	REACH	Gi0/0

```
R2#show ipv6 neighbor
```

<b>IPv6 Address</b>	<b>Age</b>	<b>Link-layer Addr</b>	<b>State</b>	<b>Interface</b>
FE80::C801:9FF:FE6D:8	0	ca01.096d.0008	REACH	Gi0/0
2001:DB8::12:3456	0	ca01.096d.0008	REACH	Gi0/0

## Neighbour Discovery Protocol

- Another function of NDP allows hosts to automatically discover routers on the local network
- 2 messages are used for this process
  - Router Solicitation (RS)
    - ICMPv6 type 133
    - Send to multicast address FF00::2 (all routers)
    - Asks all routers on the local link (local network) to identify themselves
    - Sent when an interface is enabled/host is connected to the network
  - Router Advertisement (RA)
    - ICMPv6 type 134
    - Sent to multicast address FF02::1 (all nodes)
    - The router announces its presence, as well as other info about the link
    - These messages are sent in response to RS messages
    - They are also sent periodically, even if the router hasn't received an RS



## SLAAC

- Stateless Address Auto-Configuration
- Hosts use the RS/RA messages to learn the IPv6 prefix of the local link (i.e. 2001:db8::/64) and then automatically generate an IPv6 address
- Using the 'eui-64' command, need to manually config the IPv6 address
- Using 'ipv6 address autoconfig' command, you don't need to enter the prefix. The device uses NDP to learn the prefix used on the local link
- The device will use EUI-64 to generate the interface ID, or it will be randomly generated (depending on the device/maker)

```
R2(config)#int g0/0
R2(config-if)#ipv6 address autoconfig
R2(config-if)#do show ipv6 interface brief
GigabitEthernet0/0      [up/up]
  FE80::EF8:22FF:FE56:A600
  2001:DB8::EF8:22FF:FE56:A600
GigabitEthernet0/1      [administratively down/down]
  unassigned
GigabitEthernet0/2      [administratively down/down]
  unassigned
GigabitEthernet0/3      [administratively down/down]
  unassigned
```

## Duplicate Address Detection (DAD)

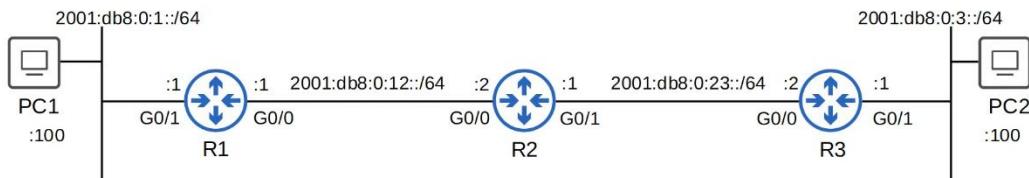
- DAD allows hosts to check if other devices on the local link are using the same IPv6 address
- Any time an IPv6-enabled interface initializes ('no shutdown' command), or an IPv6 address is configured on an interface (by any method: manual, SLAAC etc), it performs DAD
- DAD uses 2 messages (learnt earlier)
  - NS
  - NA
- The host will send an NS to its own IPv6 address

- If it doesn't get a reply, it knows the address is unique
- If it gets a reply, it means another host on the network is already using the address

```
*Oct 31 11:28:48.318: %IPV6_ND-4-DUPLICATE: Duplicate address 2001:DB8::1 on GigabitEthernet0/0
```

## IPv6 Routing

- Same as IPv4 routing
- However, the 2 processes are separate, and their routing tables are also separate
- IPv4 routing is enabled by default, IPv6 is not
  - Must be enabled with 'ipv6 unicast-routing'
  - If disabled, able to send and receive IPv6 traffic, but cannot route them (will not forward btw networks)



```
R1#show ipv6 route
IPv6 Routing Table - default - 5 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
      B - BGP, HA - Home Agent, MR - Mobile Router, R - RIP
      H - NHRP, I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
      IS - ISIS summary, D - EIGRP, EX - EIGRP external, NM - NEMO
      ND - ND Default, NDp - ND Prefix, DCE - Destination, NDr - Redirect
      RL - RPL, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1
      OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
      la - LISP alt, lr - LISP site-registrations, ld - LISP dyn-eid
      lA - LISP awav, a - Application
C 2001:DB8:0:1::/64 [0/0]
  via GigabitEthernet0/1, directly connected
L 2001:DB8:0:1::1/128 [0/0]
  via GigabitEthernet0/1, receive
C 2001:DB8:0:12::/64 [0/0]
  via GigabitEthernet0/0, directly connected
L 2001:DB8:0:12::1/128 [0/0]
  via GigabitEthernet0/0, receive
L FF00::/8 [0/0]
  via Null0, receive
```

- A connected network route and local host route are auto added for each address configured on the router
- Routes for link-local addresses are not added to the routing table

## Commands

"**ipv6 route destination/prefix-length {next-hop | exit-interface [next-hop]} [ad]**"

- Directly attached static route
  - Only the exit interface is specified
  - Cannot work on Ethernet interface (e.g. Serial can)
    - Will still accept the command, but won't work
  - R1(config)# **ipv6 route 2001:db8:0:3::/64 s0/0**
- Recursive static route
  - Only the next hop is specified
  - Called recursive as has to look up the routing table again to find the interface the next hop address is connected to
  - R1(config)# **ipv6 route 2001:db8:0:3::/64 2001:db8:0:12::2**
- Fully specified
  - Both the exit interface and the next hop specified
  - R1(config)# **ipv6 route 2001:db8:0:3::/64 g0/0 2001:db8:0:12::2**

## Network route

- R1(config)# **ipv6 route 2001:db8:0:3::/64 2001:db8:0:12::2**

## Host route

- R2(config)# **ipv6 route 2001:db8:0:1::100/128 2001:db8:0:12::1**
- R2(config)# **ipv6 route 2001:db8:0:3::100/128 2001:db8:0:23::2**

## Default route

- R3(config)# **ipv6 route ::/0 2001:db8:0:23::1**

## Link-local

- Need to use fully specified static route

- Router don't know which interface the link-local address is connected to

## Access Control Lists (ACLs)

### Standard ACL

Things covered

- What are ACLs?
- ACL logic
- ACL types
- Standard numbered ACLs
- Standard named ACLs

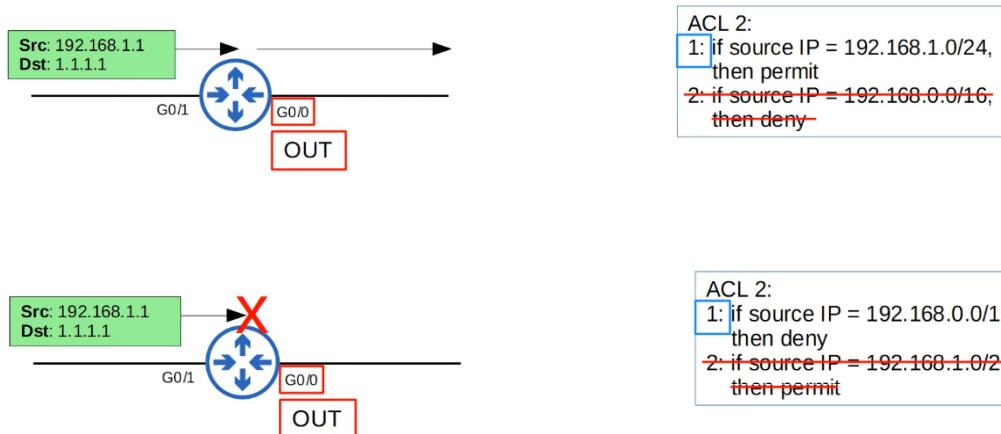
What are ACLs

- Access Control Lists
- Have multiple uses
- For now, focus from a security perspective
- ACLs function as a packet filter, instructing the router to permit or discard specific traffic
- ACLs can filter based on
  - source/destination IP addresses
  - Source/destination Layer 4 ports
  - Etc

How ACLs work

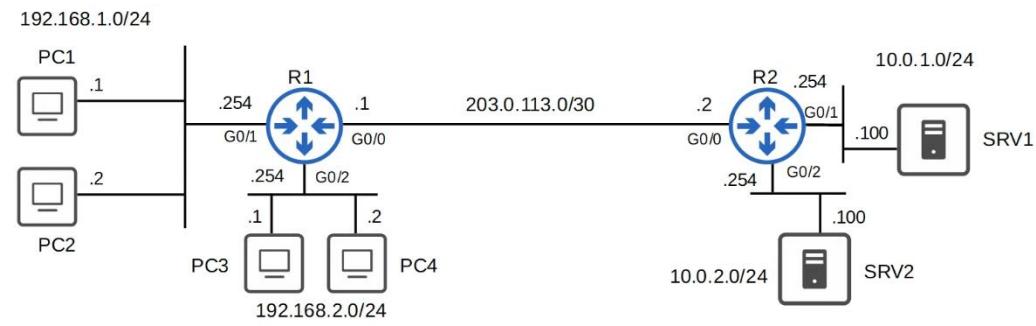
- ACLs are configured globally on the router (global config mode)
- They are an ordered sequence of ACEs (Access Control Entries)

- Configuring an ACL in global config mode will not make the ACL take effect
- The ACL must be applied to an interface
- ACLs are applied either inbound or outbound
  - Inbound - check when traffic entering interface
  - Outbound - check when traffic exiting the interface
- When the router checks a packet against the ACL, it processes the ACEs in order, from top to bottom
- If packet matches one of the ACEs in the ACL, router takes action and stop processing the ACL. All entries below the matching entry will be ignored

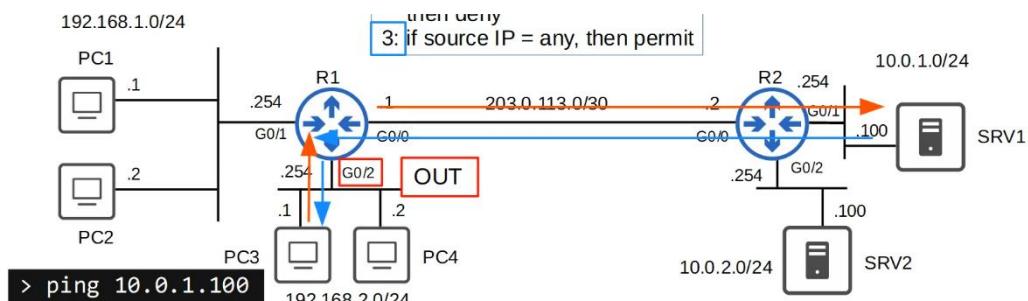


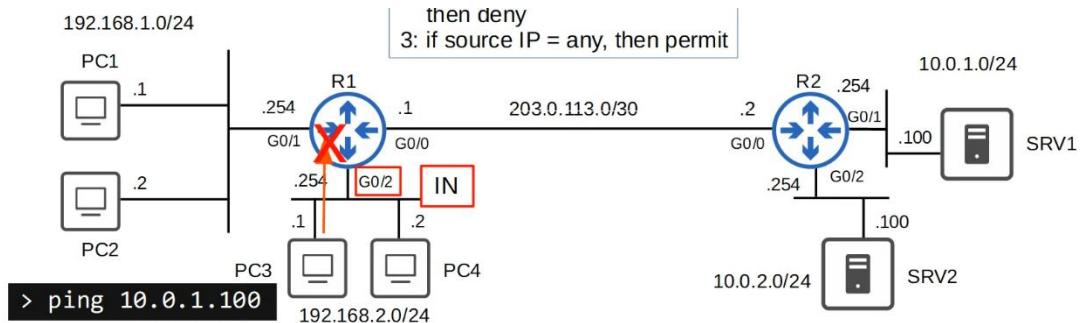
- Note: A max of 1 ACL per direction can be applied to a single interface
  - 1 interface - 1 inbound, 1 outbound
- Implicit deny
  - If traffic does not match any of the ACEs, it will be dropped
  - There is an 'implicit deny' at the end of all ACLs

Example



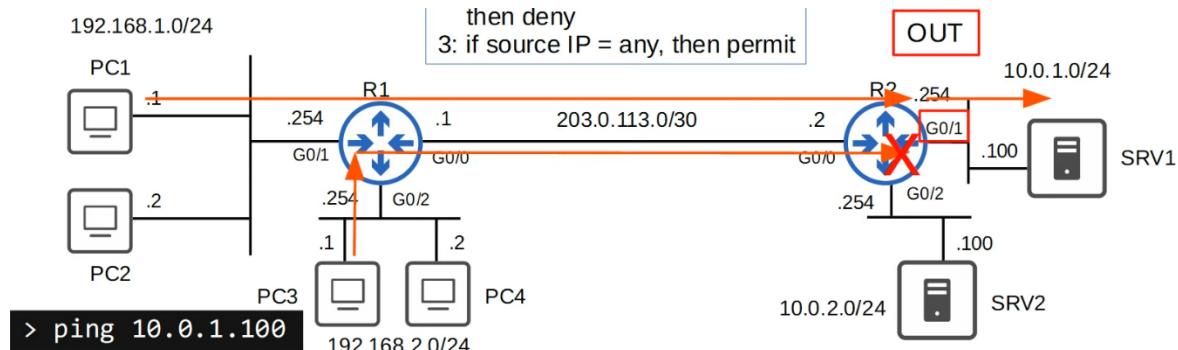
- Requirements
  - Hosts in 192.168.1.0/24 can access the 10.0.1.0/24 network
  - Hosts in 192.168.2.0/24 cannot access the 10.0.1.0/24 network
- ACE of ACL 1:
  - If source IP = 192.168.1.0/24, permit
  - If source IP = 192.168.2.0/24, deny
  - If source IP = any, permit





- Applying ACL inbound for R1 G0/2 may not be the best
  - PC3 can only communicate within its subnet
  - PC3 cannot communicate with PC1 and its subnet

### Best Method



### ACL Types

- Standard ACLs
  - Match based on Source IP address only
  - Types
    - Standard Numbered ACLs
    - Standard Named ACLs
- Extended ACLs
  - Match based on Source/Destination IP, Source/Destination Port, etc
  - Type
    - Extended Numbered ACLs
    - Extended Named ACLs

## Standard Numbered ACLs

- Match traffic based on Source IP address only
- Identified with a number (i.e. ACL 1, ACL 2)
- Different type of ACLs have a different range of numbers that can be used
  - Standard ACLs: 1-99 and 1300-1999
- Basic command to config a standard numbered ACL
  - R1(config)# **access-list number {deny | permit} ip-address wildcard-mask**

```
R1(config)# access-list 1 deny 1.1.1.1 0.0.0.0
```

```
R1(config)# access-list 1 deny 1.1.1.1
```

```
R1(config)# access-list 1 deny host 1.1.1.1
```

- The above 3 methods have the same effect of denying a host (/32)
  - If not /32, only the first one

```
R1(config)# access-list 1 permit any
```

```
R1(config)# access-list 1 permit 0.0.0.0 255.255.255.255
```

- The above 2 methods used to permit all other hosts

```
R1(config)# access-list 1 remark ## BLOCK BOB FROM ACCOUNTING ##
```

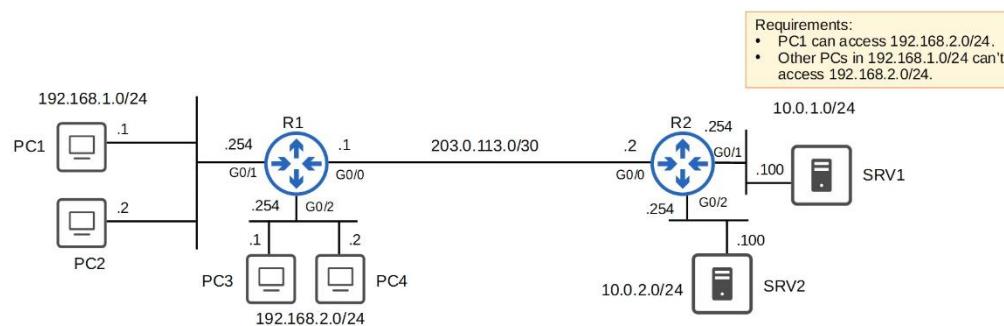
```

R1(config)#access-list 1 deny 1.1.1.1 0.0.0.0
R1(config)#access-list 1 permit 0.0.0.0 255.255.255.255
R1(config)#access-list 1 remark ## BLOCK BOB FROM ACCOUNTING ##
R1(config)#
R1(config)#do show access-lists
Standard IP access list 1
  10 deny  1.1.1.1
  20 permit any
R1(config)#
R1(config)#do show ip access-lists
Standard IP access list 1
  10 deny  1.1.1.1
  20 permit any
R1(config)#
R1(config)#do show running-config | include access-list
access-list 1 deny  1.1.1.1
access-list 1 permit any
access-list 1 remark ## BLOCK BOB FROM ACCOUNTING ##
R1(config)#

```

- Applying on interface
  - R1(config-if)# **ip access-group number {in | out}**

## Example



```

R1(config)#access-list 1 permit 192.168.1.1
R1(config)#access-list 1 deny 192.168.1.0 0.0.0.255
R1(config)#access-list 1 permit any
R1(config)#
R1(config)#interface g0/2
R1(config-if)#ip access-group 1 out
R1(config-if)#

```

Standard ACLs should be applied as close to the destination as possible.

```
R1#show access-lists
Standard IP access list 1
 10 permit 192.168.1.1
 20 deny   192.168.1.0, wildcard bits 0.0.0.255
 30 permit any
R1#
```

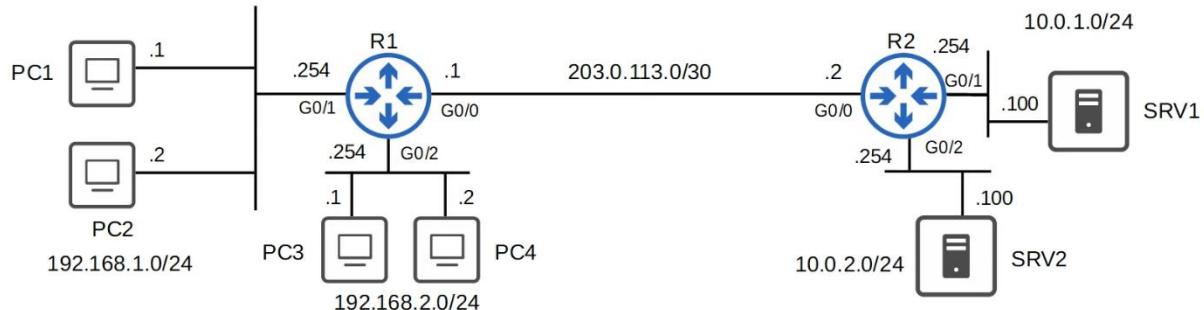
### Standard Named ACLs

- Match traffic based on source IP address only
- Identified with a name (i.e. "BLOCK\_BOB")
- Configured by entering "standard named ACL config mode", then configuring each entry within that mode
  - R1(config)# **ip access-list standard acl-name**
  - R1(config-std-nacl)# [entry-number] {**deny | permit**} ip-address wildcard-mask
- If no entry-number, it will be in interval of 10

```
R1(config)#ip access-list standard BLOCK_BOB
R1(config-std-nacl)#5 deny 1.1.1.1
R1(config-std-nacl)#10 permit any
R1(config-std-nacl)#remark ## CONFIGURED NOV 21 2020 ##
R1(config-std-nacl)#interface g0/0
R1(config-if)#ip access-group BLOCK_BOB in
```

```
R1#show access-lists
Standard IP access list BLOCK_BOB
  5 deny   1.1.1.1
 10 permit any
R1#
R1#show running-config | section access-list
ip access-list standard BLOCK_BOB
  deny   1.1.1.1
  permit any
  remark ## CONFIGURED NOV 21 2020 ##
```

## Example



### Requirements:

- PCs in 192.168.1.0/24 can't access 10.0.2.0/24.
- PC3 can't access 10.0.1.0/24.
- Other PCs in 192.168.2.0/24 can access 10.0.1.0/24.
- PC1 can access 10.0.1.0/24.
- Other PCs in 192.168.1.0/24 can't access 10.0.1.0/24.

```
R2(config)#ip access-list standard T0_10.0.2.0/24
R2(config-std-nacl)#deny 192.168.1.0 0.0.0.255
R2(config-std-nacl)#permit any
R2(config-std-nacl)#interface g0/2
R2(config-if)#ip access-group T0_10.0.2.0/24 out
R2(config-if)#
R2(config-if)#ip access-list standard T0_10.0.1.0/24
R2(config-std-nacl)#deny 192.168.2.1
R2(config-std-nacl)#permit 192.168.2.0 0.0.0.255
R2(config-std-nacl)#permit 192.168.1.1
R2(config-std-nacl)#deny 192.168.1.0 0.0.0.255
R2(config-std-nacl)#permit any
R2(config-std-nacl)#interface g0/1
R2(config-if)#ip access-group T0_10.0.1.0/24 out
R2(config-if)#

```

```
R2#show ip access-lists
Standard IP access list T0_10.0.1.0/24
 30 permit 192.168.1.1
 10 deny 192.168.2.1
 20 permit 192.168.2.0, wildcard bits 0.0.0.255
 40 deny 192.168.1.0, wildcard bits 0.0.0.255
 50 permit any
Standard IP access list T0_10.0.2.0/24
 10 deny 192.168.1.0, wildcard bits 0.0.0.255
 20 permit any
R2#

```

- Not in CCNA

- Router may reorder /32 entries
- Improves the efficiency of processing the ACL
- Does not change the effect of the ACL
- Applies to both standard named and standard numbered ACLs
- Packet Tracer does not do this

## Extended ACLs

### Things Covered

- Another way to config numbered ACLs
- Editing ACLs
- Extended numbered and named ACLs

### Configuring Numbered ACLs

- In modern IOS, can config numbered ACLs the same way as named ACLs

```
R1(config)# ip access-list standard 1
R1(config-std-nacl)# deny 192.168.1.1
R1(config-std-nacl)# permit any
```

- This is just another way
- However, in the running-config the ACL will display as if it was configured using the traditional method

```
R1(config)#ip access-list standard ?
<1-99>      Standard IP access-list number
<1300-1999>  Standard IP access-list number (expanded range)
WORD          Access-list name

R1(config)#ip access-list standard 1
R1(config-std-nacl)#deny 192.168.1.1
R1(config-std-nacl)#permit any
R1(config-std-nacl)#
R1(config-std-nacl)#do show running-config | section access-list
access-list 1 deny    192.168.1.1
access-list 1 permit any
R1(config-std-nacl)#

```

## Advantages

- Can easily delete individual entries in the ACL with 'no <entry-number>'

```
R1(config-std-nacl)#do show access-lists
Standard IP access list 1
 10 deny  192.168.1.1
 20 deny  192.168.1.2
 30 deny  192.168.3.0, wildcard bits 0.0.0.255
 40 permit any
R1(config-std-nacl)#
R1(config-std-nacl)#no 30
R1(config-std-nacl)#
R1(config-std-nacl)#do show access-lists
Standard IP access list 1
 10 deny  192.168.1.1
 20 deny  192.168.1.2
 40 permit any
R1(config-std-nacl)#

```

```
R1(config)#do show access-lists
Standard IP access list 1
 10 deny  192.168.1.1
 20 deny  192.168.1.2
 30 deny  192.168.3.0, wildcard bits 0.0.0.255
 40 permit any
R1(config)#do show running-config | section access-list
access-list 1 deny  192.168.1.1
access-list 1 deny  192.168.1.2
access-list 1 deny  192.168.3.0  0.0.0.255
access-list 1 permit any
R1(config)#no access-list 1 deny 192.168.3.0 0.0.0.255
R1(config)#do show access-lists
R1(config)#do show running-config | section access-list
R1(config)#

```

- When configuring/editing numbered ACLs from global config mode, you cannot delete individual entries, can only delete the entire ACL
- Can insert new entries in btw other entries by specifying the sequence number

```

R1(config-std-nacl)#do show access-lists
Standard IP access list 1
 10 deny  192.168.1.1
 20 deny  192.168.1.2
 40 permit any
R1(config-std-nacl)#
R1(config-std-nacl)#30 deny 192.168.2.0 0.0.0.255
R1(config-std-nacl)#
R1(config-std-nacl)#do show access-lists
Standard IP access list 1
 10 deny  192.168.1.1
 20 deny  192.168.1.2
 30 deny  192.168.2.0, wildcard bits 0.0.0.255
 40 permit any
R1(config-std-nacl)#
R1(config-std-nacl)#do show running-config | section access-list
access-list 1 deny  192.168.1.1
access-list 1 deny  192.168.1.2
access-list 1 deny  192.168.2.0 0.0.0.255
access-list 1 permit any

```

## Resequencing ACLs

- There is a resequencing function that helps edit ACLs
- R1(config)# **ip access-list resequence acl-id starting-seq-num increment**

```

R1(config)#do show access-lists
Standard IP access list 1
 1 deny  192.168.1.1
 3 deny  192.168.3.1
 2 deny  192.168.2.1
 4 deny  192.168.4.1
 5 permit any
R1(config)#
R1(config)#ip access-list resequence 1 10 10
R1(config)#
R1(config)#do show access-lists
Standard IP access list 1
 10 deny  192.168.1.1
 20 deny  192.168.3.1
 30 deny  192.168.2.1
 40 deny  192.168.4.1
 50 permit any

```

Change the sequence number of the first entry to 10.

Add 10 for every entry after that.

## Extended ACLs

- Function mostly the same as standard ACLs
- Can be numbered or named

- Numbered ACLs range: 100-199, 2000-2699
- Processed from top to bottom
- Can match traffic based on more parameters, so they are more precise (and complex) than standard ACLs
- Focus on these parameters:
  - Layer 4 protocol / port
  - Source address
  - Destination address
- Numbered ACL config
  - R1(config)# **access-list number [permit | deny] protocol src-ip dest-ip**
- Named ACL config
  - R1(config)# **ip access-list extended {name | number}**
  - R1(config-ext-acl)# **[seq-num] [permit | deny] protocol src-ip dst-ip**

## Protocol

```
R1(config)#ip access-list extended EXAMPLE
R1(config-ext-nacl)#deny ?
<0-255>  An IP protocol number
ahp          Authentication Header Protocol
eigrp        Cisco's EIGRP routing protocol
esp          Encapsulation Security Payload
gre          Cisco's GRE tunneling
icmp         Internet Control Message Protocol
igmp         Internet Gateway Message Protocol
ip           Any Internet Protocol
ipinip       IP in IP tunneling
nos          KA9Q NOS compatible IP over IP tunneling
object-group Service object group
ospf         OSPF routing protocol
pcp          Payload Compression Protocol
pim          Protocol Independent Multicast
sctp         Stream Control Transmission Protocol
tcp          Transmission Control Protocol
udp          User Datagram Protocol
```

1: ICMP
6: TCP
17: UDP
88: EIGRP
89: OSPF

## Source & Destination Address

```

R1(config-ext-nacl)#deny tcp ?
A.B.C.D      Source address
any           Any source host
host          A single source host
object-group  Source network object group

R1(config-ext-nacl)#deny tcp any ?
A.B.C.D      Destination address
any           Any destination host
eq            Match only packets on a given port number
gt            Match only packets with a greater port number
host          A single destination host
lt            Match only packets with a lower port number
neq           Match only packets not on a given port number
object-group  Destination network object group
range         Match only packets in the range of port numbers

R1(config-ext-nacl)#deny tcp any 10.0.0.0 ?
A.B.C.D      Destination wildcard bits

R1(config-ext-nacl)#deny tcp any 10.0.0.0 0.0.0.255
R1(config-ext-nacl)#

```

- In extended ACLs, to specify a /32 source or destination, have to use the "host" option or specify the wildcard mask
- Cannot just write the address without either of those like in standard ACLs
- Function of command on last line
  - Deny all packets that encapsulate a TCP segment, from any source, to destination 10.0.0.0/24

## Practice

1. Allow all traffic
<pre>R1(config-ext-nacl)#permit ip any any</pre>
2. Prevent 10.0.0.0/16 from sending UDP traffic to 192.168.1.1/32
<pre>R1(config-ext-nacl)#deny udp 10.0.0.0 0.0.255.255 host 192.168.1.1</pre>
3. Prevent 172.16.1.1/32 from pinging hosts in 192.168.0.0/24
<pre>R1(config-ext-nacl)#deny icmp host 172.16.1.1 192.168.0.0 0.0.0.255</pre>

## Matching the TCP/UDP port numbers

- When matching TCP/UDP, can optionally specify the source and/or destination port numbers to match
- Without port numbers

- R1(config-ext-nacl)# **deny tcp src-ip dest-ip**
- With port numbers
  - R1(config-ext-nacl)# **deny tcp src-ip [eq src-port-num] dest-ip [eq dst-port-num]**
- "eq 80" = equal to port 80
- "gt 80" = greater than port 80 (81 and greater)
- "lt 80" = less than port 80 (79 and less)
- "neq 80" = not equal port 80
- "range 80 100" = from port 80 to 100

```
R1(config-ext-nacl)#deny tcp any host 1.1.1.1 eq ?
<0-65535>  Port number
bgp          Border Gateway Protocol (179)
chargen     Character generator (19)
cmd          Remote commands (rcmd, 514)
daytime     Daytime (13)
discard     Discard (9)
domain      Domain Name Service (53)
drip         Dynamic Routing Information Protocol (3949)
echo         Echo (7)
exec         Exec (rsh, 512)
finger       Finger (79)
ftp          File Transfer Protocol (21)
ftp-data    FTP data connections (20)
gopher      Gopher (70)
hostname    NIC hostname server (101)
ident        Ident Protocol (113)
irc          Internet Relay Chat (194)
klogin      Kerberos login (543)
kshell      Kerberos shell (544)
login       Login (rlogin, 513)
lpd          Printer service (515)
nntp        Network News Transport Protocol (119)
onep-plain  ONEP Cleartext (15001)
onep-tls    ONEP TLS (15002)
pim-auto-rp PIM Auto-RP (496)
pop2        Post Office Protocol v2 (109)
pop3        Post Office Protocol v3 (110)
smtp        Simple Mail Transport Protocol (25)
sunrpc     Sun Remote Procedure Call (111)
tacacs     TAC Access Control System (49)
talk        Talk (517)
telnet     Telnet (23)
time        Time (37)
uucp        Unix-to-Unix Copy Program (540)
whois      Nicname (43)
www         World Wide Web (HTTP, 80)
```

```
R1(config-std-nacl)#deny tcp any host 1.1.1.1 eq 80
```

- Deny all packets destined for IP address 1.1.1.1/32, TCP port 80
- There are many more options that can be used to match (not in CCNA).

Some examples:

- "ack": match the TCP ACK flag
- "fin": match the TCP FIN flag
- "syn": match the TCP SYN flag
- "ttl": match packets with a specific TTL value
- "dscp": match packets with a specific DSCP value
- If you specify many fields, the packet must match all of the fields to match the ACL entry. Even if 1 does not match, won't match the ACL entry

## Practice

1. Allow traffic from 10.0.0.0/16 to access the server at 2.2.2.2/32 using HTTPS.

```
R1(config-ext-nacl)#permit tcp 10.0.0.0 0.0.255.255 2.2.2.2 0.0.0.0 eq 443
```

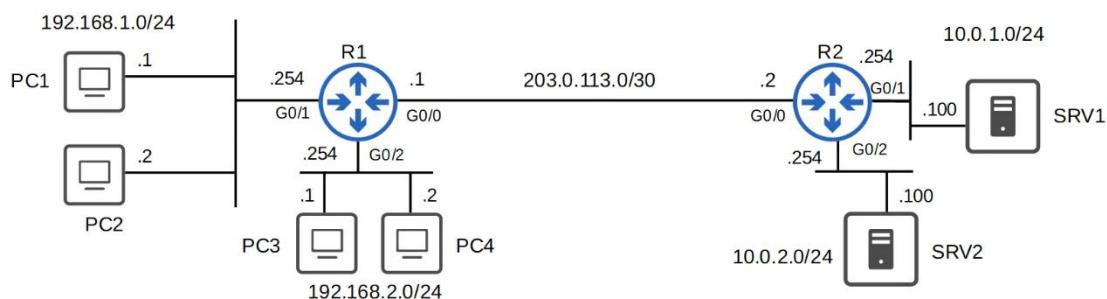
2. Prevent all hosts using source UDP port numbers from 20000 to 30000 from accessing the server at 3.3.3.3/32.

```
R1(config-ext-nacl)#deny udp any range 20000 30000 host 3.3.3.3
```

3. Allow hosts in 172.16.1.0/24 using a TCP source port greater than 9999 to access all TCP ports on server 4.4.4.4/32 except port 23.

```
R1(config-ext-nacl)#permit tcp 172.16.1.0 0.0.0.255 gt 9999 host 4.4.4.4 neq 23
```

## Example



Requirements:

- Hosts in 192.168.1.0/24 can't use HTTPS to access SRV1.
- Hosts in 192.168.2.0/24 can't access 10.0.2.0/24.
- None of the hosts in 192.168.1.0/24 or 192.168.2.0/24 can ping 10.0.1.0/24 or 10.0.2.0/24.

- NOTE: For extended ACLs, since very specific, should be applied closest to the source as possible

### Requirement 1

```
R1(config)#ip access-list extended HTTP_SRV1
R1(config-ext-nacl)#deny tcp 192.168.1.0 0.0.0.255 host 10.0.1.100 eq 443
R1(config-ext-nacl)#permit ip any any
R1(config-ext-nacl)#interface g0/1
R1(config-if)#ip access-group HTTP_SRV1 in
```

### Requirement 2

```
R1(config)#ip access-list extended BLOCK_10.0.2.0/24
R1(config-ext-nacl)#deny ip 192.168.2.0 0.0.0.255 10.0.2.0 0.0.0.255
R1(config-ext-nacl)#permit ip any any
R1(config-ext-nacl)#interface g0/2
R1(config-if)#ip access-group BLOCK_10.0.2.0/24 in
```

### Requirement 3

```
R1(config)#ip access-list extended BLOCK_ICMP
R1(config-ext-nacl)#deny icmp 192.168.1.0 0.0.0.255 10.0.1.0 0.0.0.255
R1(config-ext-nacl)#deny icmp 192.168.1.0 0.0.0.255 10.0.2.0 0.0.0.255
R1(config-ext-nacl)#deny icmp 192.168.2.0 0.0.0.255 10.0.1.0 0.0.0.255
R1(config-ext-nacl)#permit ip any any
R1(config-ext-nacl)#interface g0/0
R1(config-if)#ip access-group BLOCK_ICMP out
```

```
R1#show access-lists
Extended IP access list BLOCK_10.0.2.0/24
  10 deny ip 192.168.2.0 0.0.0.255 10.0.2.0 0.0.0.255
  20 permit ip any any
Extended IP access list BLOCK_ICMP
  10 deny icmp 192.168.1.0 0.0.0.255 10.0.1.0 0.0.0.255
  20 deny icmp 192.168.1.0 0.0.0.255 10.0.2.0 0.0.0.255
  30 deny icmp 192.168.2.0 0.0.0.255 10.0.1.0 0.0.0.255
  40 permit ip any any
Extended IP access list HTTP_SRV1
  10 deny tcp 192.168.1.0 0.0.0.255 host 10.0.1.100 eq 443
  20 permit ip any any
```

```
R1#show ip interface g0/0
GigabitEthernet0/0 is up, line protocol is up
  Internet address is 203.0.113.1/30
  Broadcast address is 255.255.255.255
  Address determined by non-volatile memory
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is BLOCK_ICMP
  Inbound access list is not set
  Proxy ARP is enabled
  Local Proxy ARP is disabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachable messages are always sent
  ICMP mask replies are never sent
```

## Layer 2 Discovery Protocols (CDP & LLDP)

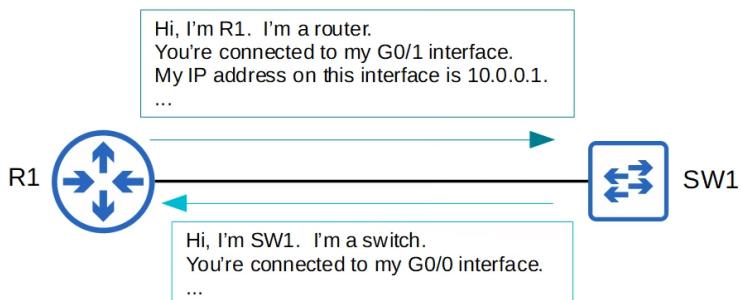
### Things covered

- Intro
- Cisco Discovery Protocol (CDP)
- Link Layer Discovery Protocol (LLDP)

### Intro

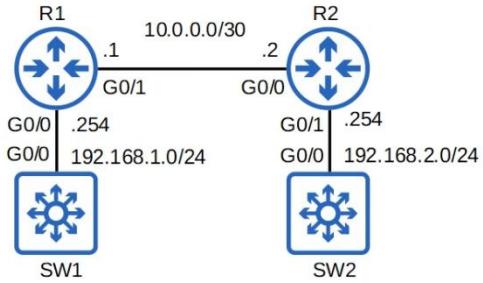
- Layer 2 Discovery Protocols such as CDP and LLDP share information with and discover info about neighbouring (connected) devices
- Operate at Layer 2
- The shared information includes: host name, IP address, device type, etc
- CDP is Cisco proprietary

- LLDP is industry standard protocol (IEEE 802.1AB)
- Because they share information about the devices in the network, they can be considered a security risk and are often not used. Up to the network engineer to decide if they want to use them or not



## CDP (Cisco Discovery Protocol)

- Cisco proprietary
- Enabled on Cisco devices by default
- CDP messages are periodically sent to multicast MAC address 0100.0CCC.CCCC
- When a device receives a CDP message, it processes and discards the message
  - Does not forward it to other devices
  - Only connected neighbours can be CDP neighbours
  - Interface must be up for device to be neighbours
- By default, CDP messages are sent every 60s
- By default, CDP hold time is 180s
  - If message not received from neighbour after 180s, neighbour removed from CDP table
- CDPv2 messages are sent by default



```

R1#show cdp
Global CDP information:
  Sending CDP packets every 60 seconds
  Sending a holdtime value of 180 seconds
  Sending CDPv2 advertisements is enabled
R1#
R1#show cdp traffic
CDP counters :
  Total packets output: 105, Input: 112
  Hdr syntax: 0, Chksum error: 0, Encaps failed: 0
  No memory: 0, Invalid packet: 0,
  CDP version 1 advertisements output: 0, Input: 0
  CDP version 2 advertisements output: 105, Input: 112
R1#
  
```

```

R1#show cdp interface
GigabitEthernet0/0 is up, line protocol is up
  Encapsulation ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
GigabitEthernet0/1 is up, line protocol is up
  Encapsulation ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
GigabitEthernet0/2 is administratively down, line protocol is down
  Encapsulation ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds
GigabitEthernet0/3 is administratively down, line protocol is down
  Encapsulation ARPA
  Sending CDP packets every 60 seconds
  Holdtime is 180 seconds

cdp enabled interfaces : 4
interfaces up          : 2
interfaces down        : 2
  
```

```

R1#show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone,
                  D - Remote, C - CVTA, M - Two-port Mac Relay

Device ID      Local Intrfce     Holdtme   Capability Platform Port ID
SW1            Gig 0/0           153        R S I       Gig 0/0
R2             Gig 0/1           146        R B         Gig 0/0

Total cdp entries displayed : 2
R1#
  
```

```
R1#show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone
Device ID      Local Intrfce     Holdtme     Capability Platform Port ID
R2             Gig 0/1          124           R            C2900    Gig 0/0
SW1            Gig 0/0          126           S            2960     Gig 0/1
```

```
R1#show cdp neighbors detail
-----
Device ID: SW1
Entry address(es):
Platform: Cisco , Capabilities: Router Switch IGMP
Interface: GigabitEthernet0/0, Port ID (outgoing port): GigabitEthernet0/0
Holdtime : 174 sec

Version :
Cisco IOS Software, vios_12 Software (vios_12-ADVENTERPRISEK9-M), Version 15.2(4.0.55)E, TEST ENGINEERING ESTG_WEEKLY BUILD, synced to END_OF_FLO_ISP
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2015 by Cisco Systems, Inc.
Compiled Tue 28-Jul-15 18:52 by sasyamal

advertisement version: 2
VIP Management Domain: ''
Native VLAN: 1
Duplex: full

-----
Device ID: R2
Entry address(es):
IP address: 10.0.0.2
Platform: Cisco , Capabilities: Router Source-Route-Bridge
Interface: GigabitEthernet0/1, Port ID (outgoing port): GigabitEthernet0/0
Holdtime : 163 sec

Version :
Cisco IOS Software, IOSv Software (VIOS-ADVENTERPRISEK9-M), Version 15.6(2)T, RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2016 by Cisco Systems, Inc.
Compiled Tue 22-Mar-16 16:19 by prod_rel_team

advertisement version: 2
Duplex: full
Management address(es):
IP address: 10.0.0.2

Total cdp entries displayed : 2
```

## CDP show commands summary

- **R1# show cdp**
  - shows basic information about CDP (timers, version)
- **R1# show cdp traffic**
  - displays how many CDP messages have been sent and received
- **R1# show cdp interface**
  - displays which interfaces CDP is enabled on
- **R1# show cdp neighbors**
  - lists CDP neighbours and some basic information about each neighbour

- R1# **show cdp neighbors detail**
  - lists each CDP neighbour with more detailed information
- R1# **show cdp entry name**
  - displays the same info as above, but for the specified neighbour only

## Config

- CDP is globally enabled by default
- CDP also enabled on each interface by default
- Enable/Disable CDP globally
  - R1(config)# **[no] cdp run**
- Enable/disable on specific interface
  - R1(config-if)# **[no] cdp enable**
- Configure timer
  - R1(config)# **cdp timer seconds**
- Configure holdtime
  - R1(config)# **cdp holdtime seconds**
- Enable/disable CDPv2 (if disable, v1 used)
  - R1(config)# **[no] cdp advertise-v2**

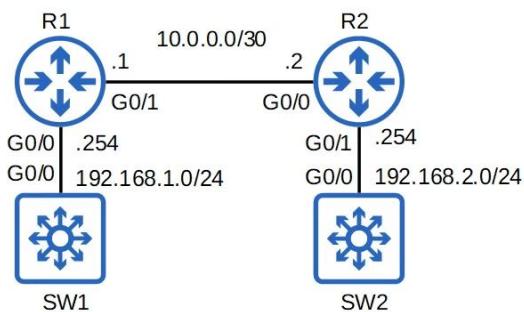
## Link Layer Discovery Protocol

- Industry standard protocol (IEEE 802.1AB)
- Usually disabled on Cisco devices by default
- Device can run both CDP and LLDP at the same time
- LLDP messages are periodically sent to multicast address 0180.C200.000E
- When device receives LLDP message, it processes and discards the message
  - Does not forward it to others
- By default, message sent every 30s

- By default, LLDP hold time is 120s
- Additional timer called 'reinitialization timer'
  - If LLDP is enabled (globally/interface), this timer will delay the actual initialization of LLDP
  - 2s by default

## Config

- LLDP globally disabled by default
- LLDP disabled on interface by default
- Enable globally
  - R1(config)# **lldp run**
- Enable on interface
  - R1(config-if)# **lldp transmit**
  - R1(config-if)# **lldp receive**
- Config timer
  - R1(config)# **lldp timer seconds**
- Config hold time
  - R1(config)# **lldp holddate seconds**
- Config re-init timer
  - R1(config)# **lldp reinit seconds**



```
R1#show lldp

Global LLDP Information:
  Status: ACTIVE
  LLDP advertisements are sent every 30 seconds
  LLDP hold time advertised is 120 seconds
  LLDP interface reinitialisation delay is 2 seconds
```

```
R1#show lldp traffic

LLDP traffic statistics:
  Total frames out: 4
  Total entries aged: 0
  Total frames in: 3
  Total frames received in error: 0
  Total frames discarded: 0
  Total TLVs discarded: 0
  Total TLVs unrecognized: 0

R1#
R1#show lldp interface

GigabitEthernet0/0:
  Tx: enabled
  Rx: enabled
  Tx state: IDLE
  Rx state: WAIT FOR FRAME

GigabitEthernet0/1:
  Tx: enabled
  Rx: enabled
  Tx state: IDLE
  Rx state: WAIT FOR FRAME

GigabitEthernet0/2:
  Tx: enabled
  Rx: enabled
  Tx state: INIT
  Rx state: WAIT PORT OPER

GigabitEthernet0/3:
  Tx: enabled
  Rx: enabled
  Tx state: INIT
  Rx state: WAIT PORT OPER
```

```
R1#show lldp neighbors

Capability codes:
  (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
  (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other

Device ID          Local Intf   Hold-time  Capability      Port ID
SW1               Gi0/0        120          R              Gi0/0
R2               Gi0/1        120          R              Gi0/0

Total entries displayed: 2
```

```
R1#show lldp neighbors detail
-----
Local Intf: Gi0/0
Chassis id: 0c04.41d2.1a00
Port id: Gi0/0
Port Description: GigabitEthernet0/0
System Name: SW1

System Description:
Cisco IOS Software, vios_12 Software (vios_12-ADVENTERPRISEK9-M), Version 15.2(4.0.55)E, TEST ENGINEERING ESTG_WEEKLY BUILD, synced to END_OF_FLO_ISP
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2015 by Cisco Systems, Inc.
Compi

Time remaining: 99 seconds
System Capabilities: B,R
Enabled Capabilities - not advertised
Management Addresses - not advertised
Auto Negotiation - not supported
Physical media capabilities - not advertised
Media Attachment Unit type - not advertised
Vlan ID: - not advertised

-----
Local Intf: Gi0/1
Chassis id: 0c04.418d.a400
Port id: Gi0/1
Port Description: GigabitEthernet0/0
System Name: R2

System Description:
Cisco IOS Software, IOSv Software (VIOS-ADVENTERPRISEK9-M), Version 15.6(2)T, RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2016 by Cisco Systems, Inc.
Compiled Tue 22-Mar-16 16:19 by prod_rel_team

Time remaining: 92 seconds
System Capabilities: B,R
Enabled Capabilities: R
Management Addresses:
IP: 10.0.0.2
```

```
R1#show lldp entry SW1
-----
Capability codes:
(R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
(W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other
-----
Local Intf: Gi0/0
Chassis id: 0c04.41d2.1a00
Port id: Gi0/0
Port Description: GigabitEthernet0/0
System Name: SW1

System Description:
Cisco IOS Software, vios_12 Software (vios_12-ADVENTERPRISEK9-M), Version 15.2(4.0.55)E, TEST ENGINEERING ESTG_WEEKLY BUILD, synced to END_OF_FLO_ISP
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2015 by Cisco Systems, Inc.
Compi

Time remaining: 119 seconds
System Capabilities: B,R
Enabled Capabilities: R
Management Addresses - not advertised
Auto Negotiation - not supported
Physical media capabilities - not advertised
Media Attachment Unit type - not advertised
Vlan ID: - not advertised
```

## LLDP show commands summary

- **R1# show lldp**
  - shows basic information about LLDP (timers, version)
- **R1# show lldp traffic**
  - displays how many LLDP messages have been sent and received

- R1# **show lldp interface**
  - displays which interfaces LLDP tx/rx is enabled on
- R1# **show lldp neighbors**
  - lists LLDP neighbors and some basic information about each neighbor
- R1# **show lldp neighbors detail**
  - lists each LLDP neighbor with more detailed information
- R1# **show lldp entry *name***
  - displays the same info as above, but for the specified neighbor only

Wireshark Capture

```
> Frame 12: 369 bytes on wire (2952 bits), 369 bytes captured (2952 bits) on interface -, id 0
  ▼ IEEE 802.3 Ethernet
    > Destination: CDP/VTP/DTP/PAgP/UDLD (01:00:0c:cc:cc:cc)
    > Source: 0c:04:41:47:57:00 (0c:04:41:47:57:00)
    Length: 355
  > Logical-Link Control
  ▼ Cisco Discovery Protocol
    Version: 2
    TTL: 180 seconds
      Checksum: 0xee0f [correct]
      [Checksum Status: Good]
    > Device ID: R1
    > Software Version
    > Platform: Cisco
    > Addresses
    > Port ID: GigabitEthernet0/0
  ▼ Capabilities
    Type: Capabilities (0x0004)
    Length: 8
    ▼ Capabilities: 0x00000005
      ..... .... ..... .... ..... .1 = Router: Yes
      ..... .... ..... .... ..... ..0. = Transparent Bridge: No
      ..... .... ..... .... ..... .1.. = Source Route Bridge: Yes
      ..... .... ..... .... ..... 0... = Switch: No
      ..... .... ..... .... ..... 0.... = Host: No
      ..... .... ..... .... ..... .0. .... = IGMP capable: No
      ..... .... ..... .... ..... 0.. .... = Repeater: No
      ..... .... ..... .... ..... 0... .... = VoIP Phone: No
      ..... .... ..... .... ..... 0.... .... = Remotely Managed Device: No
      ..... .... ..... .... ..... 0. .... .... = CVTA/STP Dispute Resolution/Cisco VT Camera: No
      ..... .... ..... .... ..... 0... .... .... = Two Port Mac Relay: No
    > IP Prefixes: 1
    > Duplex: Full
    > Management Addresses
```

```

> Frame 466: 325 bytes on wire (2600 bits), 325 bytes captured (2600 bits) on interface - . id 0
> Ethernet II, Src: 0c:04:41:d2:1a:00 (0c:04:41:d2:1a:00), Dst: LLDP_Multicast (01:80:c2:00:00:0e)
  Link Layer Discovery Protocol
    > Chassis Subtype = MAC address, Id: 0c:04:41:d2:1a:00
    > Port Subtype = Interface name, Id: Gi0/0
    > Time To Live = 120 sec
    > System Name = SW1
    > [truncated]System Description = Cisco IOS Software, vios_12 Software (vios_12-ADVENTERPRISEK9-M), Versic
    > Port Description = GigabitEthernet0/0
    > Capabilities
      0000 111. .... .... = TLV Type: System Capabilities (7)
      .... ...0 0000 0100 = TLV Length: 4
    > Capabilities: 0x0014
      .... .... .... ...0 = Other: Not capable
      .... .... .... ..0. = Repeater: Not capable
      .... .... .... .1.. = Bridge: Capable
      .... .... .... 0... = WLAN access point: Not capable
      .... .... ..1 .... = Router: Capable
      .... .... ..0. .... = Telephone: Not capable
      .... .... .0.. .... = DOCSIS cable device: Not capable
      .... .... 0... .... = Station only: Not capable
    > Enabled Capabilities: 0x0010
      .... .... .... ...0 = Other: Not capable
      .... .... .... ..0. = Repeater: Not capable
      .... .... .... .0.. = Bridge: Not capable
      .... .... .... 0... = WLAN access point: Not capable
      .... .... ..1 .... = Router: Capable
      .... .... ..0. .... = Telephone: Not capable
      .... .... .0.. .... = DOCSIS cable device: Not capable
      .... .... 0... .... = Station only: Not capable
  > End of LLDPDU

```

## Network Time Protocol (NTP)

### Things covered

- Why is time important for network devices
- Manual time config
- NTP basics
- NTP config

### The importance of time

- All devices have an internal clock (routers, switches, PC, etc)
- In Cisco can view time
  - "show clock [detail]"

```
R1#show clock
*00:16:00.857 UTC Sat Dec 26 2020
```

The default time zone is UTC  
(Coordinated Universal Time).

```
R1#show clock detail
*00:19:49.411 UTC Sat Dec 26 2020
Time source is hardware calendar
```

\* = time is not considered authoritative  
The hardware calendar is the default time source.

- The internal hardware clock will drift over time, so it is not the ideal time source
- For CCNA, the most important reason to have accurate time on a device is to have accurate logs for troubleshooting
  - Syslog

### "show logging"

```
R2#show logging
!output abbreviated!
*Dec 27 00:50:20.005: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.122.192 on GigabitEthernet0/0 from LOADING to FULL,
Loading Done
*Dec 27 01:06:38.653: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.0.6 on GigabitEthernet0/1 from LOADING to FULL,
Loading Done
*Dec 27 01:07:07.311: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.0.6 on GigabitEthernet0/1 from LOADING to FULL,
Loading Done
*Dec 27 01:08:29.924: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.0.6 on GigabitEthernet0/1 from FULL to DOWN, Neighbor
Down: Dead timer expired
*Dec 27 01:09:10.714: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.0.6 on GigabitEthernet0/1 from LOADING to FULL,
Loading Done

R2#show clock
*01:17:06.706 UTC Sun Dec 27 2020

R3#show logging
!output abbreviated!
May 23 16:24:17.320: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.0.5 on GigabitEthernet0/0 from LOADING to FULL, Loading
Done
May 23 16:25:08.758: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.0.5 on GigabitEthernet0/0 from FULL to DOWN, Neighbor
Down: Interface down or detached
May 23 16:25:10.714: %LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to down
May 23 16:25:11.716: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to down
May 23 16:26:14.976: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up
May 23 16:26:15.977: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up
May 23 16:26:20.618: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.0.5 on GigabitEthernet0/0 from LOADING to FULL, Loading
Done

R3#show clock
16:30:37.020 UTC Fri May 23 2008
```

### Manual Time Config

```
R2#clock set ?
hh:mm:ss Current Time

R2#clock set 14:30:00 ?
<1-31> Day of the month
MONTH Month of the year

R2#clock set 14:30:00 27 ?
MONTH Month of the year

R2#clock set 14:30:00 27 Dec ?
<1993-2035> Year

R2#clock set 14:30:00 27 Dec 2020 ?
<cr>

R2#clock set 14:30:00 27 Dec 2020
R2#show clock detail
14:30:05.887 UTC Sun Dec 27 2020
Time source is user configuration
```

- R1# **clock set hh:mm:ss day month year**
- Although the hardware calendar (built-in clock) is the default time-source, the hardware clock and the software clock are separate and can be configured separately

### Hardware Clock (Calendar) Config

```
R2#calendar set 14:35:00 ?
<1-31> Day of the month
MONTH Month of the year

R2#calendar set 14:35:00 27 ?
MONTH Month of the year

R2#calendar set 14:35:00 27 Dec ?
<1993-2035> Year

R2#calendar set 14:35:00 27 Dec 2020 ?
<cr>

R2#calendar set 14:35:00 27 Dec 2020
R2#show calendar
14:35:07 UTC Sun Dec 27 2020
```

- R1# **calendar set hh:mm:ss day month year**
- Typically, you want to sync 'clock' and 'calendar'
- '**clock update-calendar
- '**clock read-calendar****

```
R2#show clock
14:38:14.301 UTC Sun Dec 27 2020
R2#show calendar
00:00:03 UTC Sun Dec 27 2020
R2#clock update-calendar
R2#show clock
14:38:22.181 UTC Sun Dec 27 2020
R2#show calendar
14:38:23 UTC Sun Dec 27 2020
```

```
R2#show clock
00:00:15.788 UTC Mon Sep 6 1993
R2#show calendar
14:55:07 UTC Sun Dec 27 2020
R2#clock read-calendar
R2#show clock
14:55:12.522 UTC Sun Dec 27 2020
R2#show calendar
14:55:15 UTC Sun Dec 27 2020
```

### Config Time Zone

```

R2(config)#do show clock
15:13:33.985 UTC Sun Dec 27 2020
R2(config)#clock timezone ?
WORD name of time zone

R2(config)#clock timezone JST ?
<-23 - 23> Hours offset from UTC

R2(config)#clock timezone JST 9 ?
<0-59> Minutes offset from UTC
<cr>

R2(config)#clock timezone JST 9
R2(config)#do show clock
00:13:45.414 JST Mon Dec 28 2020
R2(config)#do clock set 15:15:00 Dec 27 2020
R2(config)#do show clock
15:15:02.129 JST Sun Dec 27 2020

```

- R1(config)# **clock timezone** *time-zone-name UTC-offset*

## Daylight Saving Time

```

R2(config)#clock summer-time ?
WORD name of time zone in summer
R2(config)#clock summer-time EDT ?
date Configure absolute summer time
recurring Configure recurring summer time
R2(config)#clock summer-time EDT recurring ?
<1-4> Week number to start
first First week of the month
last Last week of the month
<cr>
R2(config)#clock summer-time EDT recurring 2 ?
DAY Weekday to start
R2(config)#clock summer-time EDT recurring 2 Sunday ?
MONTH Month to start
R2(config)#clock summer-time EDT recurring 2 Sunday March ?
hh:mm Time to start (hh:mm)
R2(config)#clock summer-time EDT recurring 2 Sunday March 02:00 ?
<1-4> Week number to end
first First week of the month
last Last week of the month
R2(config)#clock summer-time EDT recurring 2 Sunday March 02:00 1 ?
DAY Weekday to end
R2(config)#clock summer-time EDT recurring 2 Sunday March 02:00 1 Sunday ?
MONTH Month to end
R2(config)#${-time EDT recurring 2 Sunday March 02:00 1 Sunday November ?
hh:mm Time to end (hh:mm)
R2(config)#${- recurring 2 Sunday March 02:00 1 Sunday November 02:00 ?
<1-1440> Offset to add in minutes
<cr>
R2(config)#${- recurring 2 Sunday March 02:00 1 Sunday November 02:00

```

	Canada	Northern America	Northern	Second Sunday March at 02:00 local standard time (for most of Canada)	First Sunday November at 02:00 local daylight saving time (for most of Canada)
--	--------	------------------	----------	---	--

- '\$': too long to be shown in one line

## NTP (Network Time Protocol)

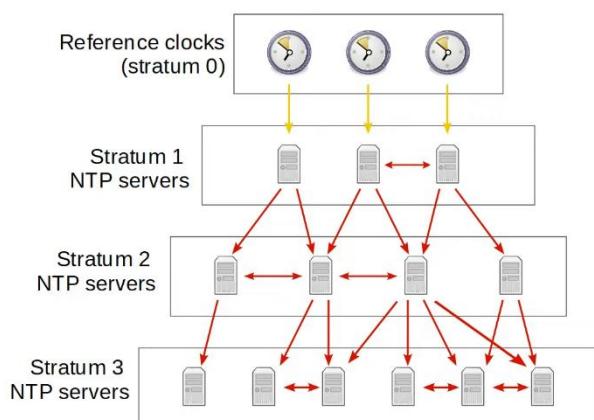
- Manually configuring the time is not scalable
- The manually configured clocks will drift, resulting in inaccurate time
- NTP allows automatic syncing of time over network
- NTP clients request the time from NTP servers
- A device can be an NTP server and client at the same time
  - Within ~1ms if NTP within same LAN

- Within ~50ms if NTP connected over a WAN/ the Internet
- Some NTP servers are 'better' than others
  - 'Distance' of an NTP server from the original **reference clock** is called **stratum**
- UDP port 123

## Reference Clock

- A reference clock is usually a very accurate time device like an atomic clock or a GPS clock
- Reference clocks are **stratum 0** within the NTP hierarchy
- NTP servers directly connected to reference clocks are **stratum 1**

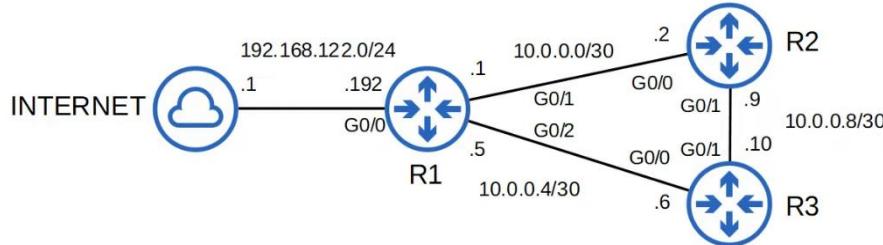
## NTP Hierarchy



- Reference clocks are stratum 0
- Stratum 1 NTP servers get time from reference clocks
- Stratum 2 NTP servers get time from stratum 1 servers
- Continue on the same
- Max: Stratum 15
  - Anything more is considered unreliable
- Devices can also 'peer' with devices at the same stratum to provide more accurate time
  - Can also act as a backup if lose access to lower stratum server
  - Called 'symmetric active' mode
- Cisco devices can operate in 3 NTP modes
  - Server
  - Client
  - Symmetric Active
- Can be all 3 at the same time
- An NTP client can sync to multiple NTP servers
- Primary Servers
  - NTP servers which get their time directly from reference clocks
  - Stratum 1 servers
- Secondary servers
  - NTP servers that get their time from other NTP servers
  - Stratum 2 or more

- Operate in server and client mode at the same time

## NTP Config



```
C:\Users\user>nslookup time.google.com
Server: dns.google
Address: 8.8.8.8

Non-authoritative answer:
Name: time.google.com
Addresses: 2001:4860:4806:::
2001:4860:4806:c:::
2001:4860:4806:8:::
2001:4860:4806:4:::
216.239.35.12
216.239.35.8
216.239.35.4
216.239.35.0
```

```
R1(config)#ntp server 216.239.35.0
R1(config)#ntp server 216.239.35.4
R1(config)#ntp server 216.239.35.8
R1(config)#ntp server 216.239.35.12
```

- Good to use multiple servers in case 1 fails
- Device will choose the one with the fastest reply
- Can put 'prefer' at the end of command if want to prefer a particular server

```
R1#show ntp associations
address          ref clock      st  when   poll  reach  delay  offset  disp
*~216.239.35.0  .GOOG.        1   43     64    17  62.007 1401.54  0.918
+~216.239.35.8  .GOOG.        1   43     64    17  64.220 1416.65  0.939
+~216.239.35.4  .GOOG.        1   47     64    17  57.669 1402.11  0.916
+~216.239.35.12 .GOOG.        1   39     64    17  62.229 1409.03  0.960
* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
```

- '\*': currently connected to the server
- '+': candidate, but not currently syncing time to that server
- '~': configured, means user key in
- '-/x': device will not sync to that server

```
R1#show ntp status
Clock is synchronized, stratum 2, reference is 216.239.35.12
nominal freq is 1000.0003 Hz, actual freq is 999.5003 Hz, precision is 2**14
ntp uptime is 295800 (1/100 of seconds), resolution is 1001
reference time is E393F0A9.1F758C5B (05:50:33.122 UTC Mon Dec 28 2020)
clock offset is 1343.7280 msec, root delay is 49.13 msec
root dispersion is 2275.31 msec, peer dispersion is 3.44 msec
loopfilter state is 'SPIK' (Spike), drift is 0.000499999 s/s
system poll interval is 64, last update was 173 sec ago.
```

- 'Clock is synchronized': R1 is able to sync to one of the configured servers
- 'stratum 2'
  - Because R1 is syncing its time to Google's NTP servers, it automatically becomes an NTP server itself (stratum lvl 1 higher than Google's NTP servers).
  - Now other devices can sync their time to R1
- 'reference is 216.....': refers to the NTP server IP address

```
R1(config)#do show clock detail
06:56:32.315 UTC Mon Dec 28 2020
Time source is NTP
R1(config)#do show calendar
05:23:06 UTC Mon Dec 28 2020
R1(config)#clock timezone JST 9
R1(config)#ntp update-calendar
R1(config)#do show clock detail
15:57:33.078 JST Mon Dec 28 2020
Time source is NTP
R1(config)#do show calendar
15:57:36 JST Mon Dec 28 2020
```

- The hardware clock tracks the date and time on the device even if it restarts, power is lost, etc
  - When the system is restarted, the hardware clock is used to initialize the software clock

```
R1(config)#interface loopback0
R1(config-if)#ip address 10.1.1.1 255.255.255.255
R1(config-if)#exit
R1(config)#ntp source loopback0
```

- Use loopback address so that other devices can connect to it and when an interface on R1 is down, other devices can use another route since connect to loopback address
  - Configure dynamic routing (e.g. OSPF)

```
R2(config)#ntp server 10.1.1.1
R2(config)#do show ntp associations

  address          ref clock      st  when   poll reach  delay  offset  disp
*~10.1.1.1        216.239.35.12  2    0     64      1  7.038 -13.128 3937.5
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
R2(config)#do show ntp status
Clock is synchronized, [stratum 3, reference is 10.1.1.1]
...
```

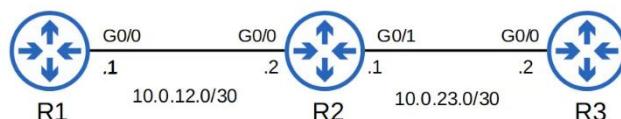
```
R3(config)#ntp server 10.1.1.1
R3(config)#ntp server 10.2.2.2
R3(config)#do show ntp associations

  address          ref clock      st  when   poll reach  delay  offset  disp
*~10.1.1.1        216.239.35.0   2    1     64      0  0.000  0.000 15937.
~10.2.2.2         10.1.1.1       3    1     64      0  0.000  0.000 15937.
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
```

- 10.2.2.2 is R2 loopback interface
- R1 10.1.1.1 is preferred due to its lower stratum level

### Config NTP server mode

```
R1(config)#ntp ?
access-group      Control NTP access
allow             Allow processing of packets
authenticate     Authenticate time sources
authentication-key Authentication key for trusted time sources
broadcastdelay   Estimated round-trip delay
clock-period     Length of hardware clock tick
logging           Enable NTP message logging
master            Act as NTP master clock
max-associations Set maximum number of associations
maxdistance       Maximum Distance for synchronization
mindistance       Minimum distance to consider for clockhop
orphan            Threshold Stratum for orphan mode
panic             Reject time updates > panic threshold (default 1000Sec)
passive           NTP passive mode
peer              Configure NTP peer
server            Configure NTP server
source            Configure interface for source address
trusted-key      Key numbers for trusted time sources
update-calendar  Periodically update calendar with NTP time
```



- R1 is not connected to any NTP server but now ones to become an NTP server

```
R1(config)#ntp master ?
<1-15> Stratum number
<cr>

R1(config)#ntp master
R1(config)#do show ntp associations
address      ref clock      st  when  poll  reach  delay  offset  disp
*~127.127.1.1 .LOCL.        7    2    16   377  0.000  0.000  0.292
* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
R1(config)#do show ntp status
Clock is synchronized, stratum 8, reference is 127.127.1.1
...
```

The default stratum of the **ntp master** command is 8.

```
R2(config)#ntp server 10.0.12.1
R2(config)#do show ntp associations
address      ref clock      st  when  poll  reach  delay  offset  disp
*~10.0.12.1  127.127.1.1  8    2    64   1    5.263  62.494 187.64
* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured

R3(config)#ntp server 10.0.12.1
R3(config)#do show ntp associations
address      ref clock      st  when  poll  reach  delay  offset  disp
*~10.0.12.1  127.127.1.1  8    45   64   17   21.534 -21.440  0.976
* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
```

## Config Symmetric Active Mode

```
R2(config)#ntp peer 10.0.23.2
R2(config)#do show ntp associations
address      ref clock      st  when  poll  reach  delay  offset  disp
*~10.0.12.1  127.127.1.1  8    60   64   17   24.040 206.682  0.987
~10.0.23.2   10.0.12.1    9    33   64   0    0.000  0.000  15937.
* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
```

```
R3(config)#ntp peer 10.0.23.1
R3(config)#do show ntp associations
address      ref clock      st  when  poll  reach  delay  offset  disp
*~10.0.12.1  127.127.1.1  8    11   64   37   12.605 -7.406 63.575
~10.0.23.1   10.0.12.1    9    1    64   0    0.000  0.000  15937.
* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
```

## Config NTP Authentication

- NTP authentication can be configured, although it is optional
- Allows NTP clients to ensure they only sync to intended servers
- To configure NTP authentication
  - Enable NTP authentication
    - "**ntp authenticate**"
  - Create the NTP authentication key(s)
    - "**ntp authentication-key key-number md5 key**"
  - Specify the trusted key(s)
    - "**ntp trusted-key key-number**"
  - Specify which key to use for the server
    - "**ntp server ip-address key key-number**"
    - Config not needed for NTP server (R1)

```
R1(config)#ntp authenticate
R1(config)#ntp authentication-key 1 md5 jeremysitlab
R1(config)#ntp trusted-key 1
```

```
R2(config)#ntp authenticate
R2(config)#ntp authentication-key 1 md5 jeremysitlab
R2(config)#ntp trusted-key 1
R2(config)#ntp server 10.0.12.1 key 1
R2(config)#ntp peer 10.0.23.2 key 1
```

```
R3(config)#ntp authenticate
R3(config)#ntp authentication-key 1 md5 jeremysitlab
R3(config)#ntp trusted-key 1
R3(config)#ntp server 10.0.12.1 key 1
R2(config)#ntp peer 10.0.23.1 key 1
```

## Summary

### Basic Configuration Commands

- R1(config)# **ntp server ip-address [prefer]**
- R1(config)# **ntp peer ip-address**
- R1(config)# **ntp update-calendar**
- R1(config)# **ntp master [stratum]**
- R1(config)# **ntp source interface**

## Basic Show Commands

- R1# show ntp associations
- R1# show ntp status

## Basic Authentication Commands

- R1(config)# ntp authenticate
- R1(config)# ntp authentication-key key-number md5 key
- R1(config)# ntp trusted-key key-number
- R1(config)# ntp server ip-address key key-number
- R1(config)# ntp peer ip-address key key-number

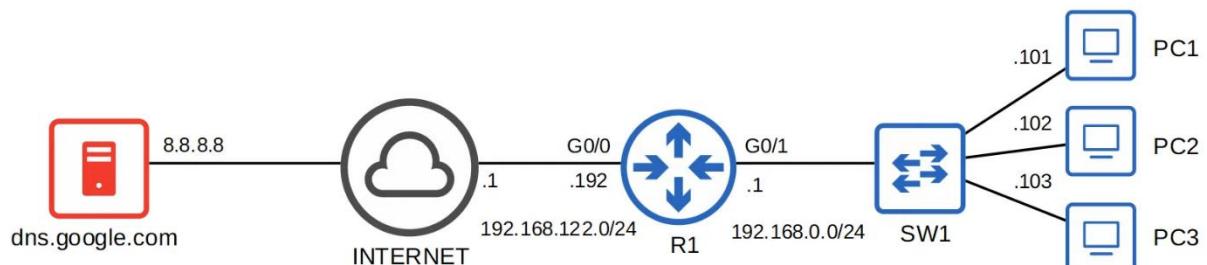
# Domain Name System (DNS)

## Things covered

- Purpose of DNS
- Basic functions of DNS
- Config DNS

## Purpose of DNS

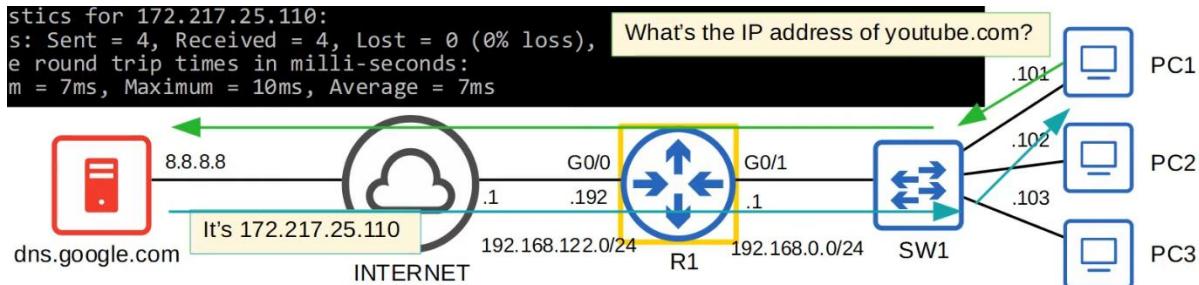
- Used to resolve human-readable names (google.com) to IP addresses
- Machines such as PCs don't use names, they use addresses (i.e. IPv4/IPv6)
- Names are much easier for us to use and remember than IP address
- When you type 'youtube.com' into a web browser, your device will ask a DNS server for the IP address of youtube.com
- The DNS server(s) your device uses can be manually configured or learned via DHCP



```
C:\Users\user>ipconfig /all  
[output omitted]  
  
Ethernet adapter ローカルエリア接続：  
  
Connection-specific DNS Suffix . . . . .  
Description . . . . . : Intel(R) 82579LM Gigabit Network Connection  
Physical Address . . . . . : 78-2B-CB-AC-08-67  
DHCP Enabled. . . . . : No  
Autoconfiguration Enabled . . . . . : Yes  
IPv4 Address. . . . . : 192.168.0.101(Preferred)  
Subnet Mask . . . . . : 255.255.255.0  
Default Gateway . . . . . : 192.168.0.1  
DNS Servers . . . . . : 8.8.8.8  
NetBIOS over Tcpip. . . . . : Enabled  
  
[output omitted]
```

```
C:\Users\user>nslookup youtube.com  
Server: dns.google  
Address: 8.8.8.8  
  
Non-authoritative answer:  
Name: youtube.com  
Addresses: 2404:6800:4004:819::200e  
          172.217.25.110  
  
C:\Users\user>ping youtube.com  
  
Pinging youtube.com [172.217.25.110] with 32 bytes of data:  
Reply from 172.217.25.110: bytes=32 time=10ms TTL=117  
Reply from 172.217.25.110: bytes=32 time=7ms TTL=117  
Reply from 172.217.25.110: bytes=32 time=7ms TTL=117  
Reply from 172.217.25.110: bytes=32 time=7ms TTL=117  
  
Ping statistics for 172.217.25.110:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
    Approximate round trip times in milli-seconds:  
        Minimum = 7ms, Maximum = 10ms, Average = 7ms
```

- Don't have to use the 'nslookup' command before sending a ping.
  - If your device don't know the correct IP address, it will automatically ask the server



- R1 is not acting as a DNS server or client. It simply forwarding packets
- NO DNS configuration is required for R1

No.	Time	Source	Destination	Protocol	Length	Info
1087	08:55:44.458619	192.168.0.101	8.8.8.8	DNS	71	Standard query 0x0002 A youtube.com
1088	08:55:44.500043	8.8.8.8	192.168.0.101	DNS	87	Standard query response 0x0002 A youtube.com A 172.217.25.110
1089	08:55:44.508888	192.168.0.101	8.8.8.8	DNS	71	Standard query 0x0003 AAAA youtube.com
1115	08:55:44.641775	8.8.8.8	192.168.0.101	DNS	99	Standard query response 0x0003 AAAA youtube.com AAAA 2404:6800:4004:819::200e
> Frame 1087: 71 bytes on wire (568 bits), 71 bytes captured (568 bits) on interface \Device\NPF_{9956EC07-3774-4811-9700-C8233E7CD172}, id 0						
> Ethernet II, Src: Dell_ac:08:67 (78:2b:cb:ac:08:67), Dst: Tp-LinkT_dd:a8:e4 (98:da:c4:dd:a8:e4)						
> Internet Protocol Version 4, Src: 192.168.0.101, Dst: 8.8.8.8						
> User Datagram Protocol, Src Port: 49286, Dst Port: 53						
` Domain Name System (query)						
Transaction ID: 0x0002						
` Flags: 0x0100 Standard query						
0... .... .... = Response: Message is a query						
.000 0... .... .... = Opcode: Standard query (0)						
.... 0. .... .... = Truncated: Message is not truncated						
.... 1 .... .... = Recursion desired: Do query recursively						
.... 0. .... .... = Z: reserved (0)						
.... 0....0 .... = Non-authenticated data: Unacceptable						
Questions: 1						
Answer RRs: 0						
Authority RRs: 0						
Additional RRs: 0						
` Queries						
` youtube.com: type A, class IN						
Name: youtube.com						
[Name Length: 11]						
[Label Count: 2]						
Type: A (Host Address) (1)						
Class: IN (0x0001)						
<a href="#">[Response In: 1088]</a>						

- DNS 'A' record = Used to map names to IPv4 addresses
- DNS 'AAAA' record = Used to map names to IPv6 addresses
- Standard DNS queries/responses typically use UDP
  - TCP is used for DNS messages greater than 512 bytes
  - In either case, port 53 is used

## DNS Cache

```
C:\Users\user>ipconfig /displaydns
[output omitted]

www.youtube.com
-----
Record Name . . . . . : www.youtube.com
Record Type . . . . . : 5
Time To Live . . . . . : 98
Data Length . . . . . : 8
Section . . . . . : Answer
CNAME Record . . . . . : youtube-ui.l.google.com

[output omitted]

Record Name . . . . . : youtube-ui.l.google.com
Record Type . . . . . : 1
Time To Live . . . . . : 98
Data Length . . . . . : 4
Section . . . . . : Answer
A (Host) Record . . . . . : 172.217.25.110

[output omitted]
```

- Devices will save the DNS server's responses to a local DNS cache
- Devices don't have to query the server every single time they want to access a particular destination
- "ipconfig /displaydns"

```
C:\Users\user>ipconfig /flushdns
Windows IP Configuration

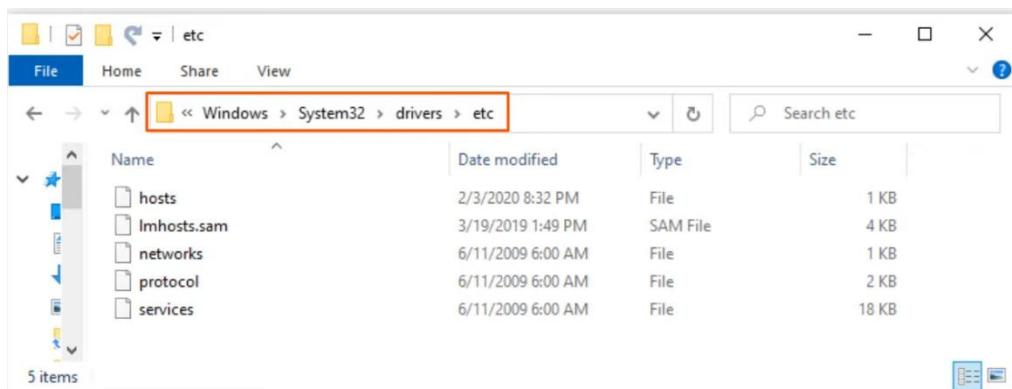
Successfully flushed the DNS Resolver Cache.

C:\Users\user>ipconfig /displaydns
Windows IP Configuration

C:\Users\user>
```

- "ipconfig /flushdns" used to clear the DNS cache

## Host file



- Most computers have a list of host files which stores a list of hosts and IP addresses

The screenshot shows two windows side-by-side. On the left is a Notepad window titled 'hosts - Notepad' containing the Windows hosts file. It includes comments about the file's purpose, examples of entries, and localhost mappings. A specific entry '192.168.0.1 R1' is highlighted with a red box. On the right is a terminal window showing the command 'ping R1' and its output. The output shows four successful replies from the IP 192.168.0.1.

```

hosts - Notepad
File Edit Format View Help
Copyright (c) 1993-2008 Microsoft Corp.

This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
This file contains the mappings of IP addresses to host names. Each
entry should be kept on an individual line. The IP address should
be placed in the first column followed by the corresponding host name.
The IP address and the host name should be separated by at least one
space.

Additionally, comments (such as these) may be inserted on individual
lines or following the machine name denoted by a '#' symbol.

For example:
# 102.54.94.97 rhino.acme.com      # source server
# 38.25.63.10  x.acme.com          # x client host

192.168.0.1 R1
# localhost name resolution is handled within DNS itself.
# 127.0.0.1      localhost
# .1           localhost

Ln 21, Col 58    100%    Windows (CRLF)    UTF-8
  
```

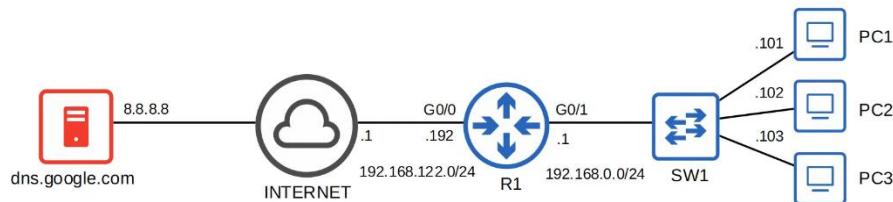
```

C:\Users\user>ping R1
Pinging R1 [192.168.0.1] with 32 bytes of data:
Reply from 192.168.0.1: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
  
```

## DNS in Cisco IOS

- For hosts in a network to use DNS, you don't need to configure DNS on the routers
  - They will simply forward the DNS messages like any other packets
- However, a Cisco router can be configured as a DNS server, although it is rare
  - If an internal (within a LAN) DNS server is used, usually it's a windows or Linux server
- A Cisco router can also be configured as a DNS client



## Set as server

R1(config)#ip dns server	Configure R1 to act as a DNS server.
R1(config)#ip host R1 192.168.0.1 R1(config)#ip host PC1 192.168.0.101 R1(config)#ip host PC2 192.168.0.102 R1(config)#ip host PC3 192.168.0.103	Configure a list of hostname/IP address mappings.
R1(config)#ip name-server 8.8.8.8	Configure a DNS server that R1 will query if the requested record isn't in its host table.
R1(config)#ip domain lookup	Enable R1 to perform DNS queries. (enabled by default) (old version of the command is <b>ip domain-lookup</b> )

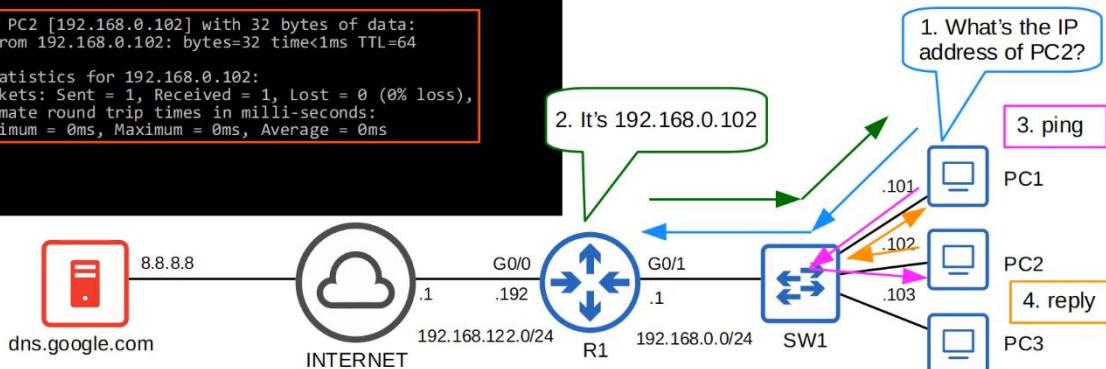
## Ping

```
C:\Users\user>ipconfig /all
[output omitted]

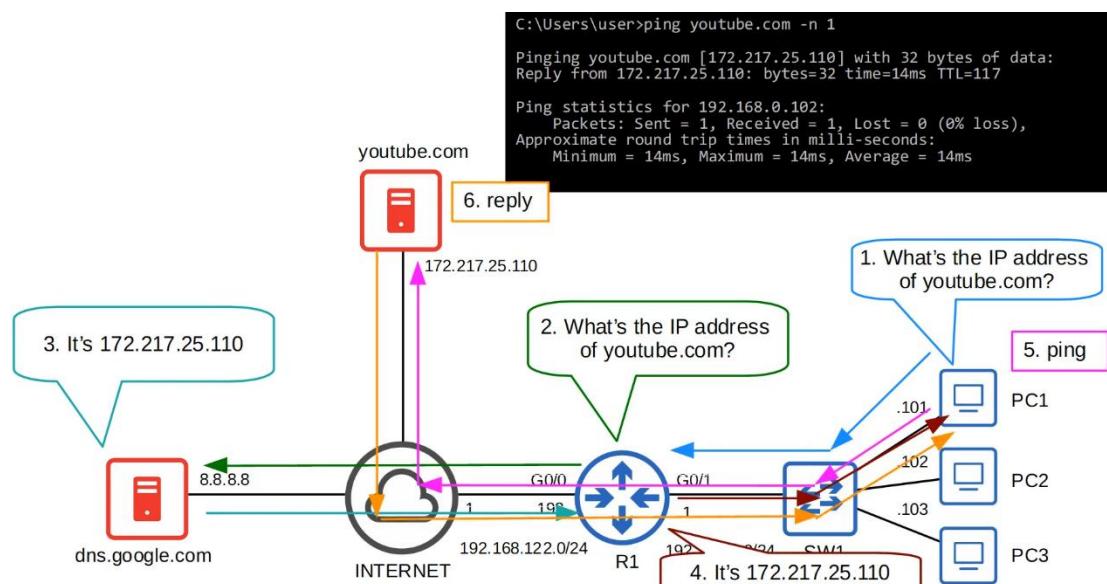
IPv4 Address . . . . . : 192.168.0.101(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.0.1
DNS Servers . . . . . : 192.168.0.1
NetBIOS over Tcpip. . . . . : Enabled

[output omitted]

C:\Users\user>ping PC2 -n 1
Pinging PC2 [192.168.0.102] with 32 bytes of data:
Reply from 192.168.0.102: bytes=32 time<1ms TTL=64
Ping statistics for 192.168.0.102:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```



- After the reply from R1 (step 2), PC1 will have the IP address of R2 in its cache table
  - Won't need to send a request anymore for IP address of R2



```

R1#show hosts
Default domain is not set
Name/address lookup uses domain service
Name servers are 8.8.8.8

Codes: UN - unknown, EX - expired, OK - OK, ?? - revalidate
      temp - temporary, perm - permanent
      NA - Not Applicable None - Not defined

Host          Port  Flags     Age  Type   Address(es)
youtube.com    None  (temp, OK)  0    IP    172.217.25.110
R1            None  (perm, OK)  4    IP    192.168.0.1
PC1           None  (perm, OK)  1    IP    192.168.0.101
PC2           None  (perm, OK)  4    IP    192.168.0.102
PC3           None  (perm, OK)  4    IP    192.168.0.103

```

## Set as client

```

R1(config)#do ping youtube.com
Translating "youtube.com"
% Unrecognized host or address, or protocol not running.

R1(config)#ip name-server 8.8.8.8 → Configure R1 to use the specified DNS server.
R1(config)#ip domain lookup → Enable R1 to perform DNS queries. (default)

R1(config)#do ping youtube.com
Translating "youtube.com"...domain server (8.8.8.8) [OK]

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.217.25.110, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/10/13 ms

R1(config)#ip domain name jeremysitlab.com

```

(optional)

- Configure the default *domain name*.
- This will be automatically appended to any hostnames without a specified *domain*.
- ie. `ping pc1` will become `ping pc1.jeremysitlab.com`
- (old version of the command: `ip domain-name`)
- I will cover this command again in a later video (about SSH).

## Review

Windows:

```
C:\Users\user>ipconfig /all  
C:\Users\user>nslookup name  
C:\Users\user>ipconfig /displaydns  
C:\Users\user>ipconfig /flushdns  
C:\Users\user>ping ip-address -n number
```

Cisco IOS:

```
R1(config)#ip dns server  
R1(config)#ip host hostname ip-address  
R1(config)#ip name-server ip-address  
R1(config)#ip domain lookup  
R1(config)#ip domain name domain-name  
R1#show hosts
```

## Dynamic Host Configuration Protocol (DHCP)

### Things Covered

- Purpose of DHCP
- Basic functions of DHCP
- Config

### Purpose of DHCP

- DHCP is used to allow hosts to automatically/dynamically learn various aspects of their network configuration (i.e. IP address, subnet mask, default gateway, DNS server) without manual/static configuration
- It is an essential part of modern networks
- Typically used for 'client devices' such as workstations (PC), phones, etc
- Devices such as routers, servers, etc are usually manually configured

- In small networks (such as home networks), the router typically acts as the DHCP server for the hosts in the LAN
- In larger networks, the DHCP server is usually a Windows/Linux server

## Basic Functions of DHCP

```
C:\Users\user>ipconfig /all
[output omitted]

Ethernet adapter Ethernet0:
  Connection-specific DNS Suffix . : This PC was previously assigned this IP address by the DHCP server,
  Description . . . . . : Intel(R) Dual Band Wireless-AC 7265
  Physical Address. . . . . : 78-2B-CB-AC-08-67
  DHCP Enabled. . . . . : Yes
  Autoconfiguration Enabled . . . . . : Yes
  IPv4 Address. . . . . : 192.168.0.167 (Preferred)
  Subnet Mask . . . . . : 255.255.255.0
  Lease Obtained. . . . . : Saturday, January 23, 2021 12:02:04 PM
  Lease Expires . . . . . : Saturday, January 23, 2021 2:02:05 PM
  Default Gateway . . . . . : 192.168.0.1
  DHCP Server . . . . . : 192.168.0.1
  DNS Servers . . . . . : 192.168.0.1
  NetBIOS over Tcpip. . . . . : Enabled
[output omitted]
```

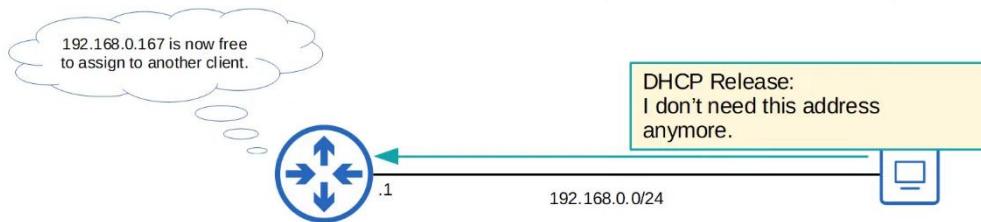
- Lease Obtained/Expires
  - DHCP server 'leases' IP addresses to clients
  - These leases are usually not permanent, and the client must give up the address at the end of the lease

## "ipconfig /release"

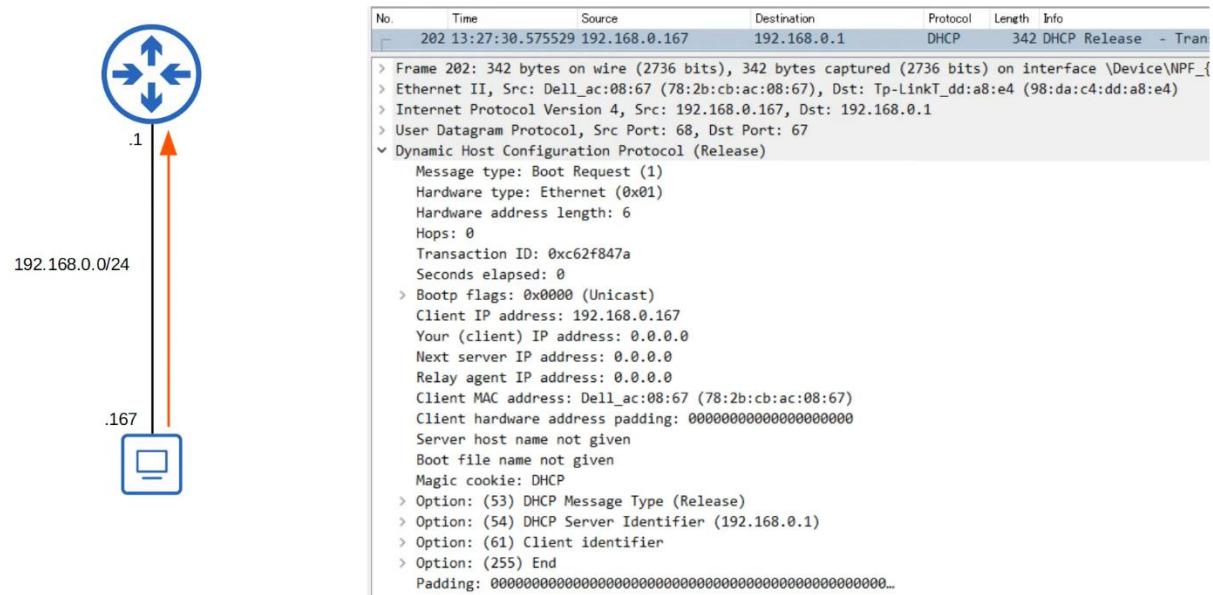
```
C:\Users\user>ipconfig /release
Windows IP Configuration
[output omitted]

Ethernet adapter Ethernet0:
  Connection-specific DNS Suffix . :
  Default Gateway . . . . . :

[output omitted]
```



## DHCP Release



- Ports
  - DHCP Server: UDP 67
  - DHCP Client: UDP 68

## Ipconfig /renew

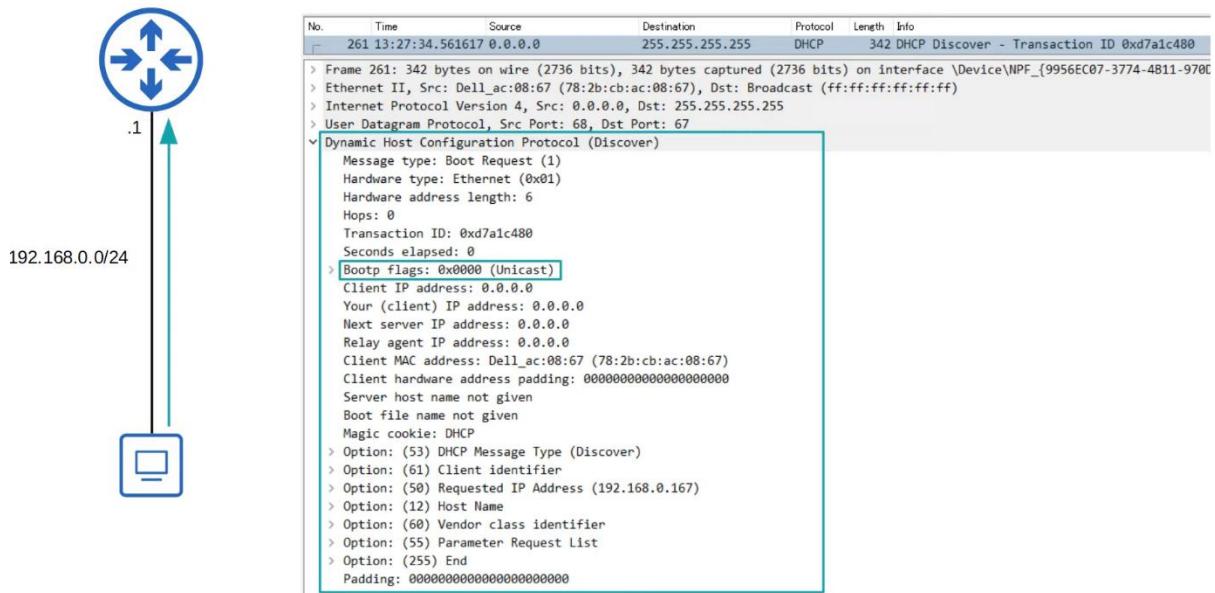
```
C:\Users\user>ipconfig /renew
C:\Users\user>ipconfig /all

Ethernet adapter Ethernet0:

  Connection-specific DNS Suffix  . : 
  Description . . . . . : Intel(R) 82579LM Gigabit Network Connection
  Physical Address . . . . . : 78-2B-CB-AC-08-67
  DHCP Enabled. . . . . : Yes
  Autoconfiguration Enabled . . . . . : Yes
  IPv4 Address. . . . . : 192.168.0.167(Preferred)
  Subnet Mask . . . . . : 255.255.255.0
  Lease Obtained. . . . . : Saturday, January 23, 2021 3:07:39 PM
  Lease Expires . . . . . : Saturday, January 23, 2021 5:07:38 PM
  Default Gateway . . . . . : 192.168.0.1
  DHCP Server . . . . . : 192.168.0.1
  DNS Servers . . . . . : 192.168.0.1
  NetBIOS over Tcpip. . . . . : Enabled
```

## DHCP Discover

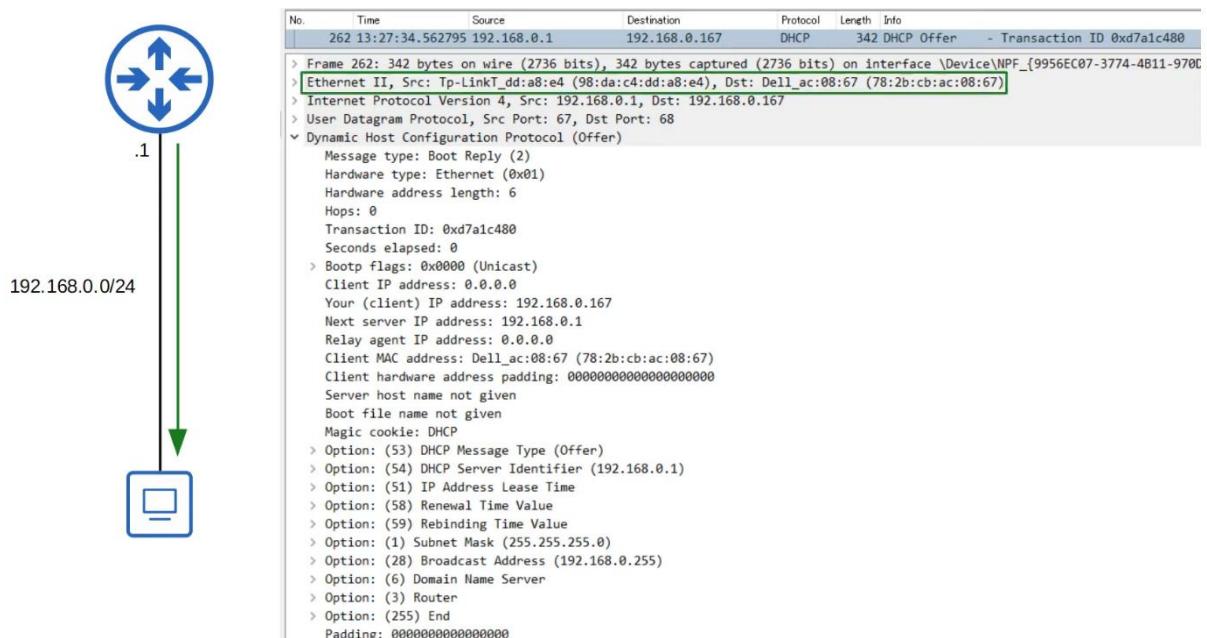
- Client send out "are there any DHCP servers in this network? I need an IP address"



- Client (PC) sends a broadcast message
- Don't have an IP, so uses 0.0.0.0
- Option (50): Requested IP address
  - The PC previously used that IP address and now requesting to use it again

## DHCP Offer

- Server to client: "How about this IP address?"

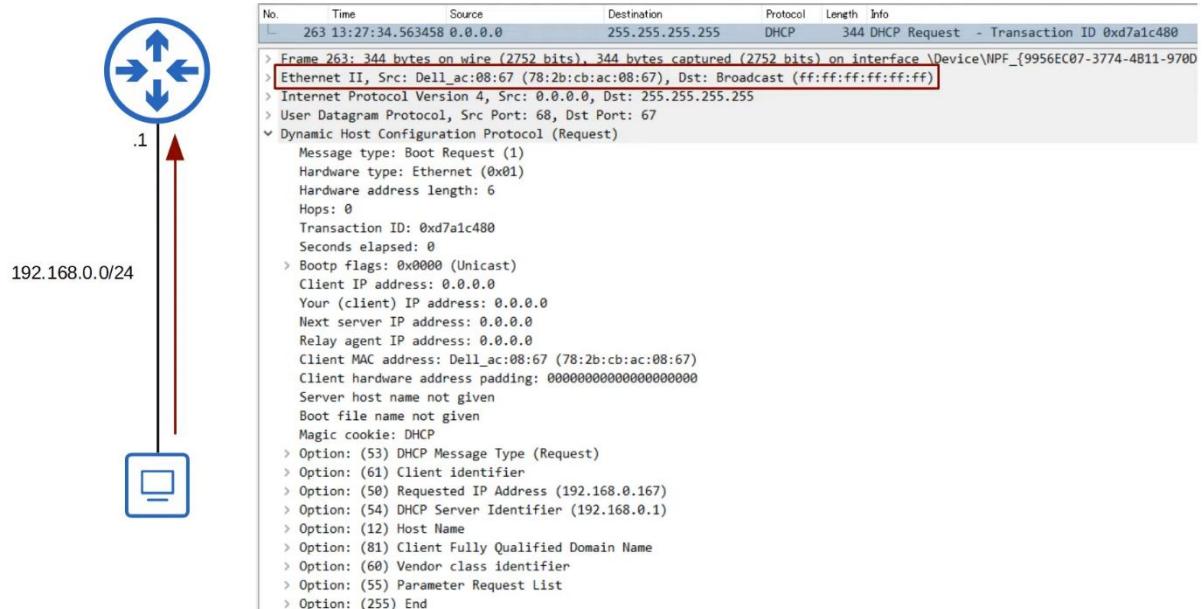


- Destination MAC: PC's MAC address

- Destination IP: Offered IP address
- Bootp flags: unicast
  - The DHCP offer message can be either unicast or broadcast
  - Requested by the PC
  - If broadcast, destination MAC is all F, and destination IP is 255.255.255.255
- The IP address of the PC is still not configured now, and some PCs would not accept unicast message as a result
  - That is why broadcast is needed

## DHCP Request

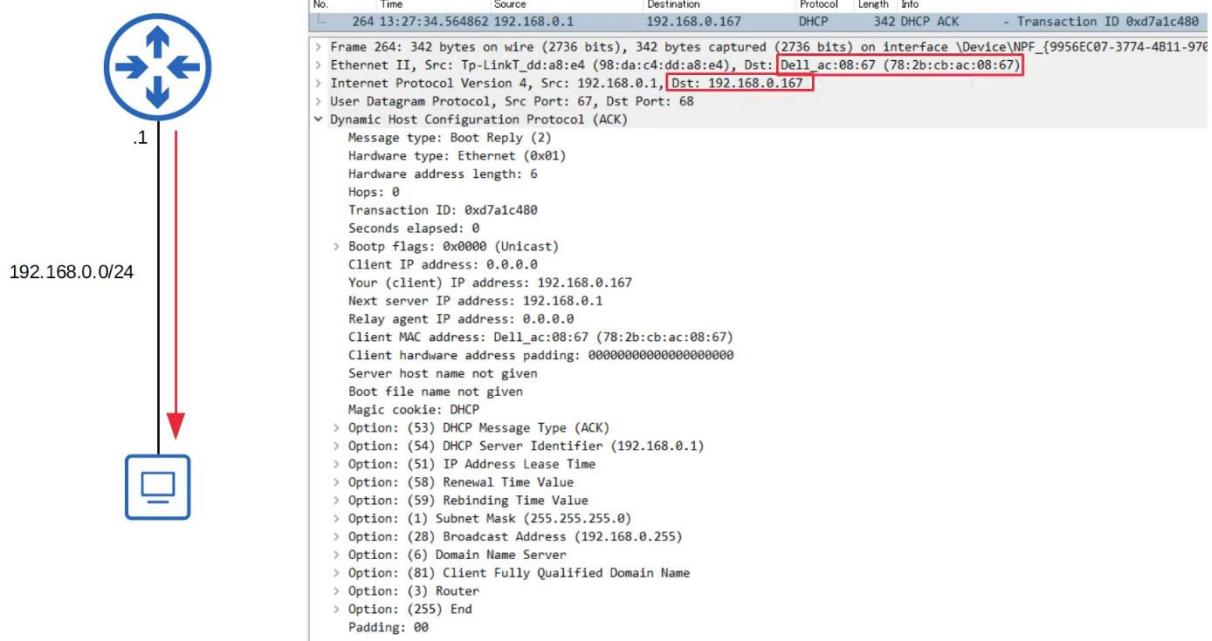
- DHCP Client to DHCP server
  - "I want to use the IP address you offered me"
- There may be multiple DHCP server, so PC may have received multiple DHCP offers
  - Client to have tell which server it wants to accept from
  - Normally accept the first offer received



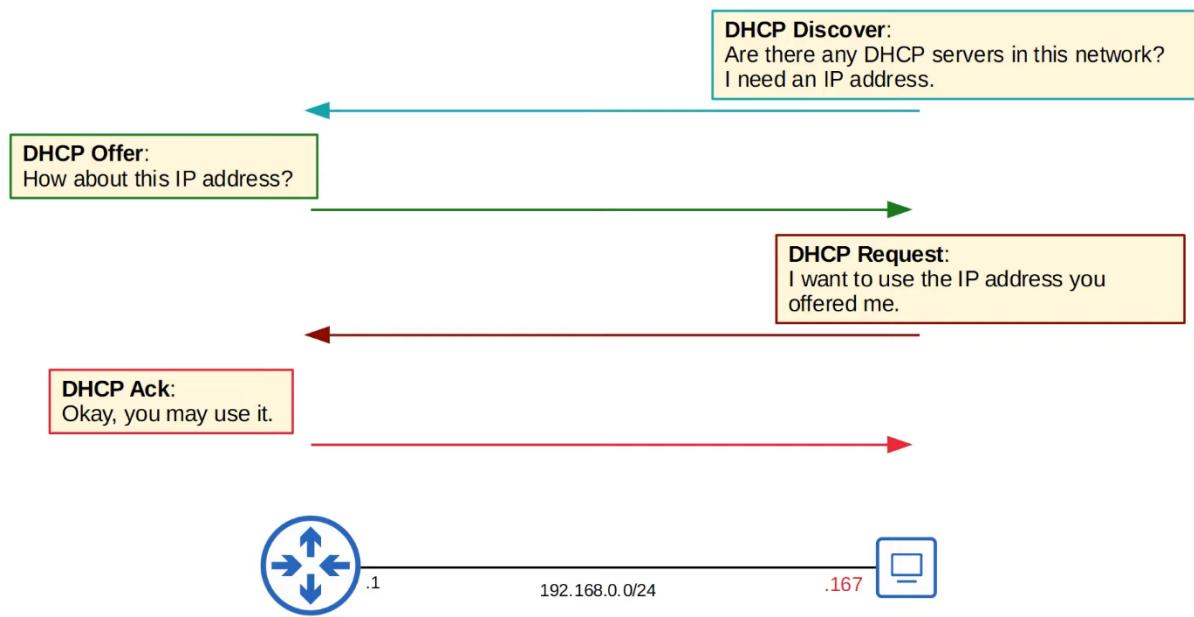
- Option (54)
  - This is how the PC specify which DHCP server to accept from

## DHCP Ack

- DHCP server -> DHCP client
  - "Ok, you may use it"
- Once client receives this message, client will configure the IP address on its interface



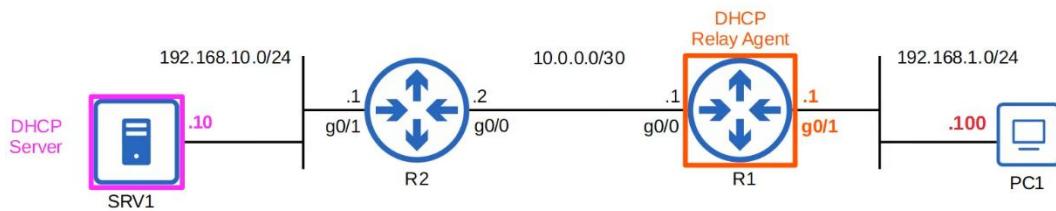
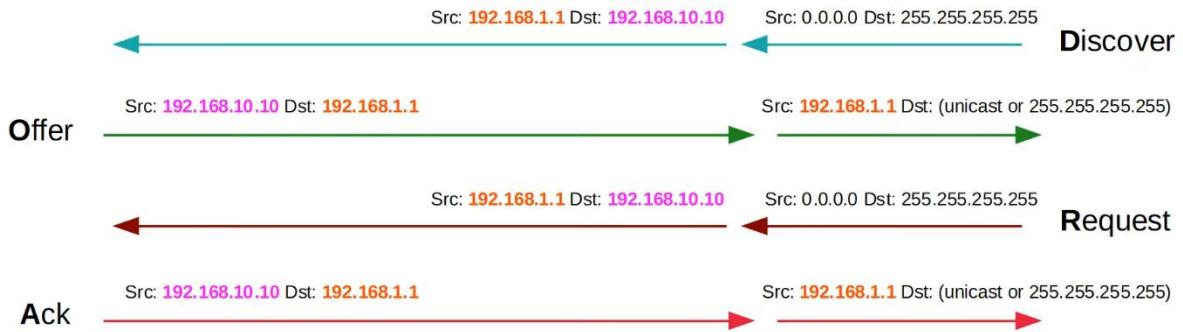
## Summary



<b>Discover</b>	Client → Server	Broadcast
<b>Offer</b>	Server → Client	Broadcast or Unicast
<b>Request</b>	Client → Server	Broadcast
<b>Ack</b>	Server → Client	Broadcast or Unicast
<b>Release</b>	Client → Server	Unicast

### DHCP Relay

- Some network engineers might choose to configure each router to act as the DHCP server for its connected LANs
- However, large enterprises often choose to use a centralized DHCP server
- If the server is centralized, it won't receive the DHCP clients' broadcast messages
- To fix this, you can configure a router to act as a DHCP relay agent
- The router will forward the clients broadcast DHCP messages to the remote DHCP server as a unicast message



## Config

### DHCP Server

```

R1(config)#ip dhcp excluded-address 192.168.1.1 192.168.1.10
Specify a range of addresses that won't be given to DHCP clients.

R1(config)#ip dhcp pool LAB_POOL
Create a DHCP pool.

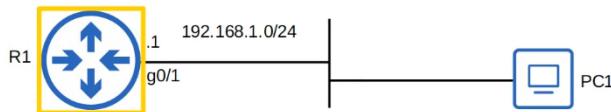
R1(dhcp-config)#network 192.168.1.0 ?
/nn or A.B.C.D Network mask or prefix length
<cr>
R1(dhcp-config)#network 192.168.1.0 /24
Specify the subnet of addresses to be assigned to clients (except the excluded addresses)

R1(dhcp-config)#dns-server 8.8.8.8
Specify the DNS server that DHCP clients should use.

R1(dhcp-config)#domain-name jeremysitlab.com
Specify the domain name of the network.
(i.e. PC1 = pc1.jeremysitlab.com)

R1(dhcp-config)#default-router 192.168.1.1
Specify the default gateway.

R1(dhcp-config)#lease 0 5 30
Specify the lease time.
lease days hours minutes OR
lease infinite
  
```



- DHCP pool

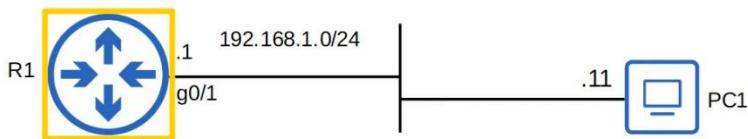
- A subnet of addresses that can be assigned to DHCP clients and other info such as DNS server and default gateway
- Should create a separate DHCP pool for each network the router is acting as a DHCP server

```
R1#show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address      Client-ID/          Lease expiration      Type
               Hardware address/
               User name
192.168.1.11    0100.0c29.e727.39     Jan 24 2021 10:52 AM  Automatic
```

```
C:\Users\user>ipconfig /all

Ethernet adapter Ethernet0:

  Connection-specific DNS Suffix  . : jeremysitlab.com
  Description . . . . . : Intel(R) PRO/1000 MT Network Connection #2
  Physical Address. . . . . : 00-0C-29-E7-27-39
  DHCP Enabled. . . . . : Yes
  Autoconfiguration Enabled . . . . . : Yes
  IPv4 Address. . . . . : 192.168.1.11(Preferred)
  Subnet Mask . . . . . : 255.255.255.0
  Lease Obtained. . . . . : Saturday, January 24, 2021 2:22:35 PM
  Lease Expires . . . . . : Saturday, January 24, 2021 7:52:35 PM
  Default Gateway . . . . . : 192.168.1.1
  DHCP Server . . . . . : 192.168.1.1
  DNS Servers . . . . . : 8.8.8.8
  NetBIOS over Tcpip. . . . . : Enabled
```



## DHCP Relay Agent

```

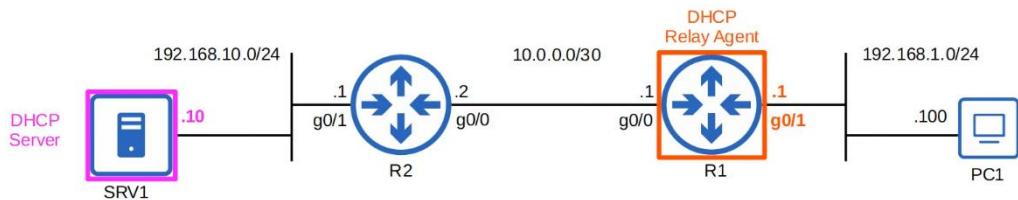
R1(config)#interface g0/1
Configure the interface connected to the subnet
of the client devices.

R1(config-if)#ip helper-address 192.168.10.10
Configure the IP address of the DHCP server
as the 'helper' address.

R1(config-if)#do show ip interface g0/1
GigabitEthernet0/1 is up, line protocol is up
  Internet address is 192.168.1.1/24
  Broadcast address is 255.255.255.255
  Address determined by non-volatile memory
  MTU is 1500 bytes
  Helper address is 192.168.10.10

[output omitted]

```



## DHCP Client

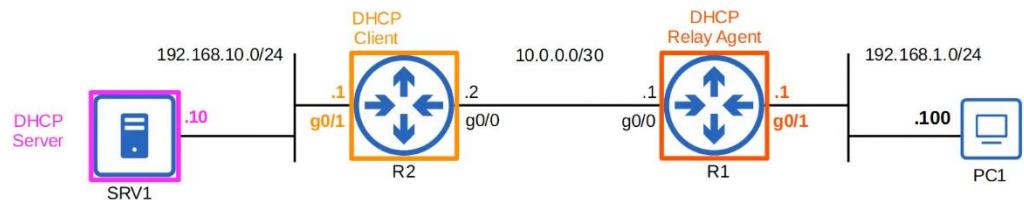
```

R2(config)#interface g0/1
R2(config-if)#ip address dhcp
R2(config-if)#do sh ip interface g0/1
GigabitEthernet0/1 is up, line protocol is up
  Internet address is 192.168.10.1/24
  Broadcast address is 255.255.255.255
  Address determined by DHCP

[output omitted]

```

Use the **ip address dhcp** mode to tell the router to use DHCP to learn its IP address.



## Summary

```
C:\Users\user> ipconfig /release  
C:\Users\user> ipconfig /renew
```

```
R1(config)# ip dhcp excluded-address Low-address high-address  
R1(config)# ip dhcp pool pool-name  
R1(dhcp-config)# network ip-address {/prefix-length | subnet-mask}  
R1(dhcp-config)# dns-server ip-address  
R1(dhcp-config)# domain-name domain-name  
R1(dhcp-config)# default-router ip-address  
R1(dhcp-config)# lease {days hours minutes | infinite}  
R1# show ip dhcp binding
```

R1(config-if)# ip helper-address ip-address      DHCP relay agent  
R1(config-if)# ip address dhcp      DHCP client

DHCP server

DHCP relay agent

DHCP client

### Simple Network Management Protocol (SNMP)

Things covered

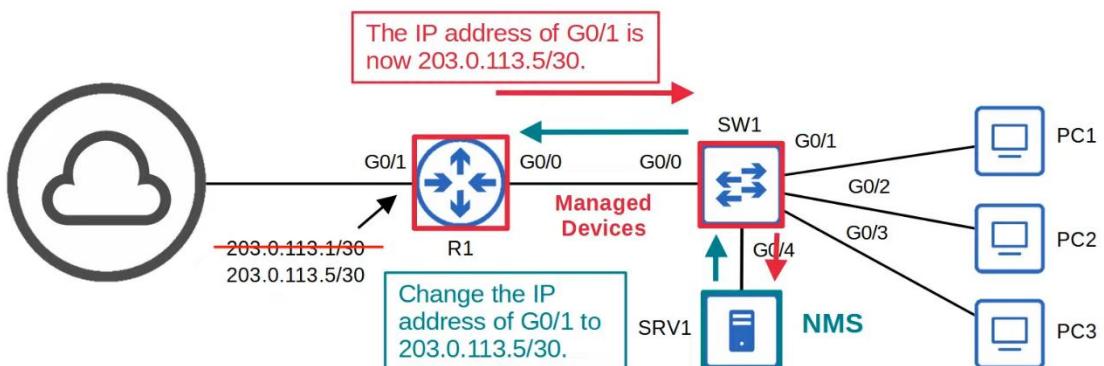
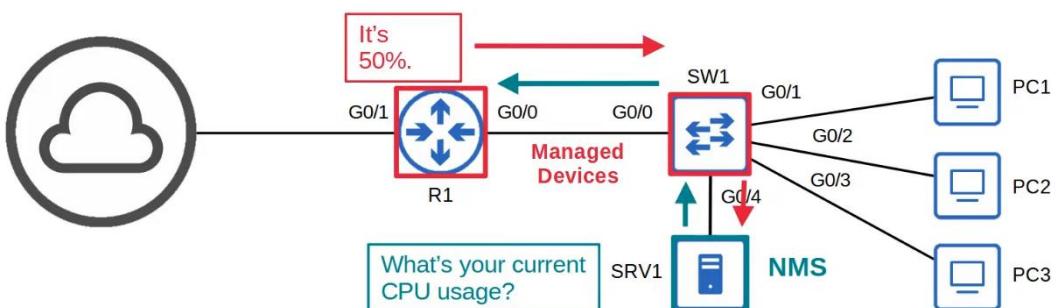
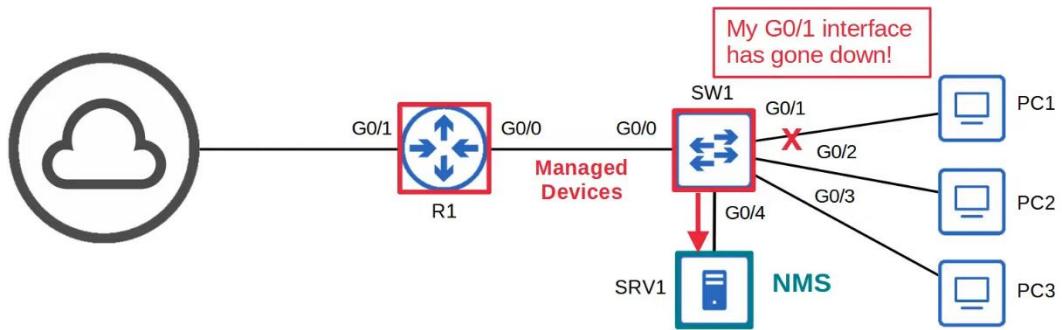
- SNMP overview
- SNMP versions
- SNMP messages
- Config

## SNMP (Simple Network Management Protocol)

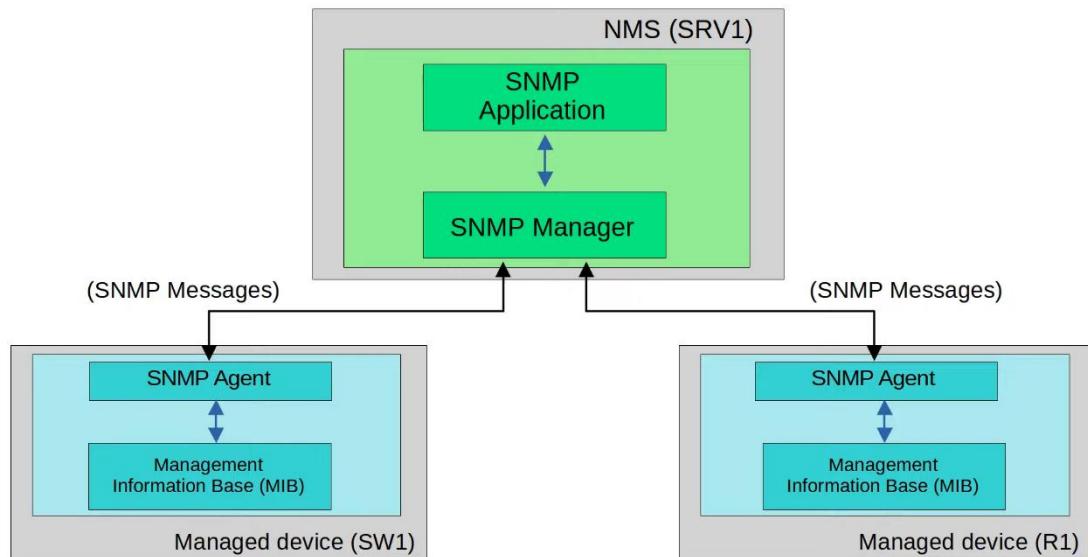
- SNMP is an industry-standard framework and protocol that was originally released in 1988
- SNMP can be used to monitor the status of devices, make configuration changes, etc
- There are 2 main type of devices in SNMP
  - Managed Devices
    - These are devices being managed using SNMP
    - E.g. Network devices like routers and switches
  - Network Management Station (NMS)
    - The device(s) managing the managed devices
    - This is the SNMP 'server'

### **SNMP Operations**

- There are 3 main operations used in SNMP
  1. Managed devices can notify the NMS of events
  2. The NMS can ask the managed devices for information about their current status
  3. The NMS can tell the managed devices to change aspects of their configuration



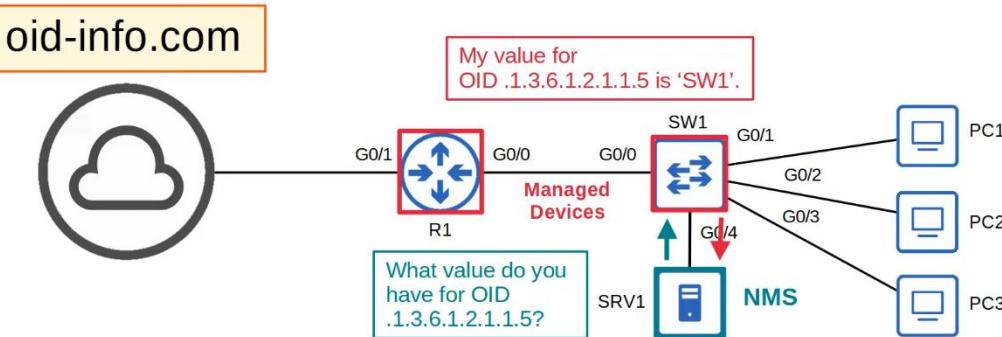
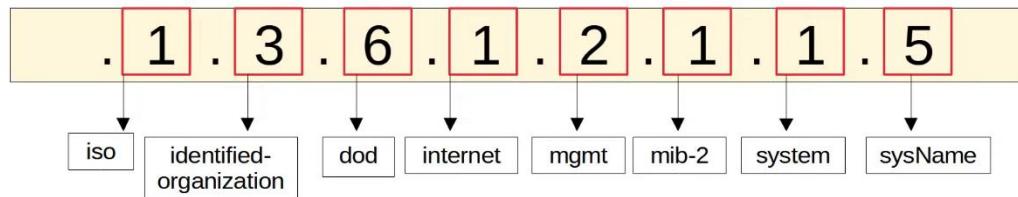
## SNMP Components



- NMS (SRV1)
  - SNMP Manager
    - The software on the NMS that interacts with the managed devices
    - It receives notifications, sends requests for information, sends configuration changes, etc
  - SNMP Application
    - Provides an interface for the network admin to interact with
    - Display alerts, statistics, charts, etc
- Managed Devices (SW1, R1)
  - SNMP Agent
    - The SNMP software running on the managed devices that interacts with the SNMP Manager on the NMS
    - It sends notifications to/receive messages from the NMS
  - Management Information Base (MIB)
    - The structure that contains the variables that are managed by SNMP
    - Each variable is identified with an object ID (OID)
    - Example variables: Interface status, traffic throughput, CPU usage, temperature, etc

## SNMP OIDs

- SNMP Object IDs are organized in a hierarchical structure



## SNMP Versions

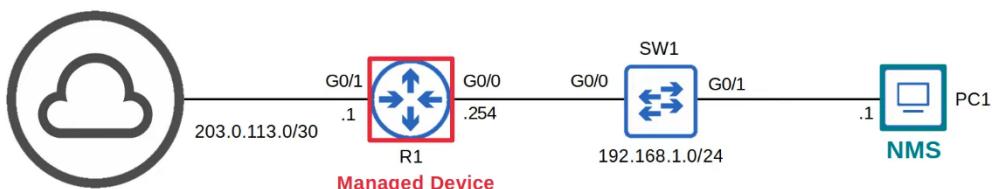
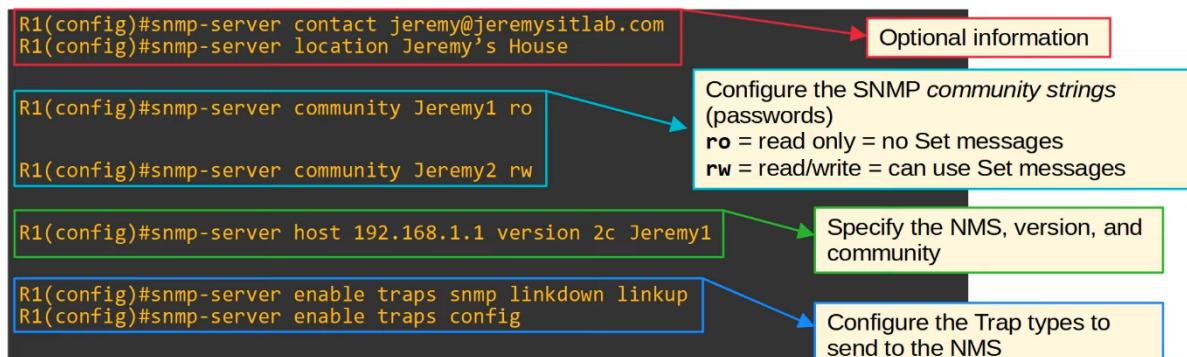
- Many versions of SNMP have been proposed/developed, however, only 3 major versions have achieved wide-spread use
- SNMPv1
  - Original version of SNMP
- SNMPv2c
  - Allow the NMS to retrieve large amounts of information in a single request, so it is more efficient
  - 'c' refers to the 'community strings' used as passwords in SNMPv1, removed in SNMPv2, and then added back for SNMPv2c
- SNMPv3
  - A much more secure version of SNMP that supports strong encryption and authentication
  - Should be used whenever possible

## SNMP Messages

Message Class	Description	Messages
Read	Messages sent by the <b>NMS</b> to read information from the <b>managed devices</b> . (ie. What's your current CPU usage %?)	<i>Get</i> <i>GetNext</i> <i>GetBulk</i>
Write	Messages sent by the <b>NMS</b> to change information on the <b>managed devices</b> . (ie. change an IP address)	<i>Set</i>
Notification	Messages sent by the <b>managed devices</b> to alert the <b>NMS</b> of a particular event. (ie. interface going down)	<i>Trap</i> <i>Inform</i>
Response	Messages sent in response to a previous message/request.	<i>Response</i>

- Read
  - Get
    - A request sent from the manager to the agent to retrieve the value of a variable (OID), or multiple variables
    - The agent will send a Response message with the current value of each variable
  - GetNext
    - A request sent from the manager to the agent to discover the available variables in the MIB
    - Gets the next element from the MIB
  - GetBulk
    - A more efficient version of the GetNext message (introduced in SNMPv2)
- Write
  - Set
    - A request sent from the manager to the agent to change the value of one or more variables
    - The agent will send a Response message with the new values
- Notification
  - Trap
    - A notification sent from the agent to the manager
    - The manager does not send a Response message to acknowledge that it received the Trap, so these messages are 'unreliable'
  - Inform
    - A notification message that is acknowledged with a Response message
    - Originally used for communications btw managers, but later updates allow agents to send inform messages to managers
  
- Port Numbers
  - SNMP Agent = UDP 161
  - SNMP Manager = UDP 162

## SNMPv2c Config



## Wireshark Capture

- Link btw R1 and internet down

No.	Time	Source	Destination	Protocol	Length	Info
209	13:55:21.662570	192.168.1.254	192.168.1.1	SNMP	221	221 snmpV2-trap 1.3.6.1.2.1.1.
> Frame 209: 221 bytes on wire (1768 bits), 221 bytes captured (1768 bits) on interface -, id 0						
> Ethernet II, Src: 0c:1c:1a:87:fb:00 (0c:1c:1a:87:fb:00), Dst: 0c:1c:1a:50:80:01 (0c:1c:1a:50:80:01)						
> Internet Protocol Version 4, Src: 192.168.1.254, Dst: 192.168.1.1						
> User Datagram Protocol, Src Port: 65385, Dst Port: 162						
Simple Network Management Protocol						
version: v2c (1)						
community: Jeremy1						
data: snmpV2-trap (7)						
snmpV2-trap						
request-id: 14						
error-status: noError (0)						
error-index: 0						
variable-bindings: 6 items						
> 1.3.6.1.2.1.1.3.0: 104924						
> 1.3.6.1.6.3.1.1.4.1.0: 1.3.6.1.6.3.1.1.5.3 (iso.3.6.1.6.3.1.1.5.3)						
> 1.3.6.1.2.1.2.2.1.1.2: 2						
> 1.3.6.1.2.1.2.2.1.2.2: 4769676162697445746865726e6574302f31						
> 1.3.6.1.2.1.2.2.1.3.2: 6						
> 1.3.6.1.4.1.9.2.2.1.1.20.2: 61646d696e6973747261746976656c7920646f776e						

15	924	1.3.6.1.6.3.1.1.5.3 (iso	OID: {iso(1) identified-organization(3) dod(6) internet(1) snmpV2(6) snmpModules(3) snmpMIB(1) snmpMIBObjects(1) snmpTraps(5) linkDown(3)}
2			1.3.6.1.6.3.1.1.5.3
476967616269744574686572			/ISO/Identified-Organization/6/1/6/3/1/1/5/3
6			

## **Summary**

- SNMP help manage devices over a network
- Managed devices
  - Devices being managed using SNMP (e.g. network devices - routers, switches)
- Network Management Stations (NMS)
  - SNMP 'servers' that manage devices
  - Receive notifications from managed devices
  - Changes settings on managed devices
  - Check status of managed devices
- Variables are stored in the Management Information Base (MIB) and identified using Object IDs (OID)
- Main versions
  - SNMPv1
  - SNMPv2c
  - SNMPv3
- SNMP messages
  - Get
  - GetNext
  - GetBulk
  - Set
  - Trap
  - Inform
  - Response

## Syslog

### Things covered

- Syslog overview
- Syslog message format
- Syslog facility and severity levels
- Config

### **Overview**

- Syslog is an industry standard for message logging
- On network devices, Syslog can be used to log events such as changes in interface status (up -> down), changes in OSPF neighbour status, system restarts etc

- The message can be displayed in the CLI, saved in the device's RAM, or sent to an external Syslog server
- Logs are essential when troubleshooting issues, examining the cause of incidents etc
- Syslog and SNMP are both used for monitoring and troubleshooting of devices
  - They are complementary, but their functionalities are different

## **Format**

**seq: time stamp: %facility-severity-MNEMONIC: description**

- Seq
  - A sequence number indicating the order/sequence of messages
- Time stamp
  - A time stamp indicating the time the message was generated
  - Seq and time stamp may/ may not be displayed based on device's config
- Facility
  - A value that indicate which process on the device generated the message
- Severity
  - A number that indicates the severity of logged event
- MNEMONIC
  - A short code for the message, indicating what happened
- Description
  - Detailed information about the event being reported

## **Severity Levels**

Level	Keyword	Description
0	<b>Emergency</b>	System is unusable
1	<b>Alert</b>	Action must be taken immediately
2	<b>Critical</b>	Critical conditions
3	<b>Error</b>	Error conditions
4	<b>Warning</b>	Warning conditions
5	<b>Notice</b>	Normal but significant condition ( <b>Notification</b> )
6	<b>Informational</b>	Informational messages
7	<b>Debugging</b>	Debug-level messages

- Level 5
  - In official RFC document is Notice
  - In Cisco CLI is Notification
- Each vendor may interpret each level differently
- Every Awesome Cisco Engineer Will Need Ice Cream Daily

## Syslog Message Examples

seq:time stamp: %facility-severity-MNEMONIC:description

```
*Feb 11 03:02:55.304: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up
```

```
*Feb 11 05:04:39.606: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.1.2 on GigabitEthernet0/0 from LOADING to FULL, Loading Done
```

```
000043: *Feb 11 05:06:43.331: %SYS-5-CONFIG_I: Configured from console by jeremy on console
```

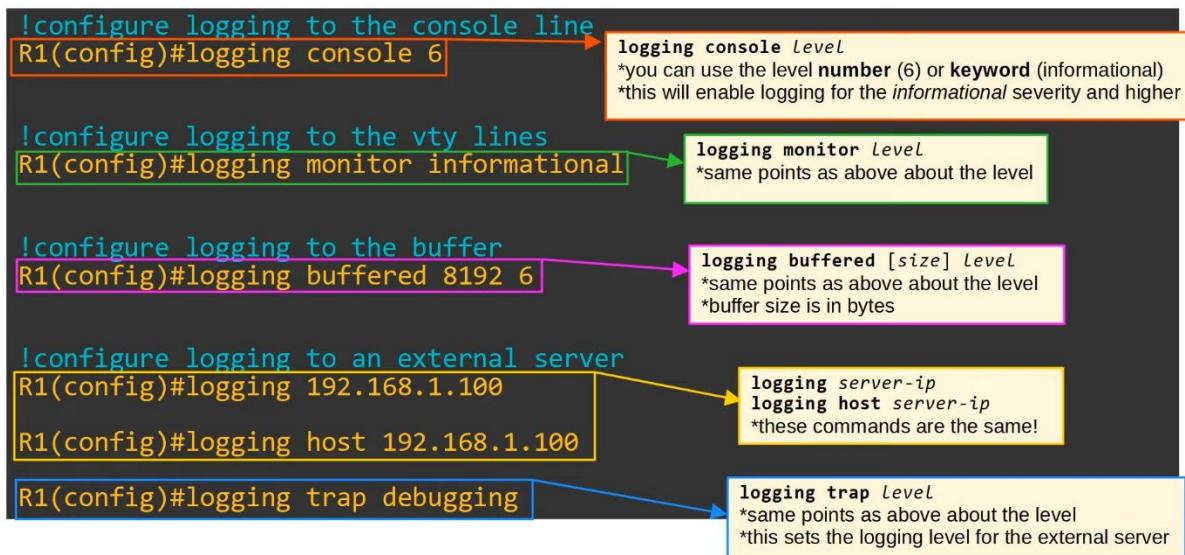
```
*Feb 11 07:27:23.346: %SYS-6-CLOCKUPDATE: System clock has been updated from 07:27:23 UTC Thu Feb 11 2021 to 16:27:23 JST Thu Feb 11 2021, configured from console by jeremy on console.
```

## Syslog Logging Locations

- Console Line

- Syslog messages will be displayed in the CLI when connected to the device via the console port
  - By default, all messages (level 0 - 7) will be displayed
- VTY Lines
  - Syslog messages will be displayed in the CLI when connected to the device via Telnet/SSH
  - Disabled by default
- Buffer
  - Syslog messages will be saved to RAM
  - By default, all messages (level 0 - 7) are displayed
  - You can view the message with "show logging"
- External server
  - Can configure the device to send Syslog messages to external server
  - Syslog servers will listen for messages on UDP port 514

## Config



## "terminal monitor"

- Even if "logging monitor level" is enabled, by default Syslog messages will not be displayed when connected via Telnet or SSH
- For the messages to be displayed, must use command
  - R1# **terminal monitor**
- This command must be used EVERY TIME you connect to the device via Telnet or SSH

## "logging synchronous"

- By default, logging messages will be displayed in the CLI while you are in the middle of typing a command
- To prevent this, should use the "logging synchronous" on the appropriate line
  - R1(config)# **line console 0**
  - R1(config-line)# **logging synchronous**
- This will cause a new line to be printed if your typing is interrupted by a message

```
R1(config)#exit
R1#show ip int
*Feb 11 09:38:41.607: %SYS-5-CONFIG_I: Configured from console by jeremy on
consoleterface brief
```

```
R1(config)#exit
R1#show ip int
*Feb 11 09:41:00.554: %SYS-5-CONFIG_I: Configured from console by jeremy on console
R1#show ip int
```

### "service timestamp" / "service sequence-numbers"

```
R1(config)#service timestamps log ?
  datetime  Timestamp with date and time
  uptime    Timestamp with system uptime
<cr>
```

**datetime** = timestamps will display the date/time when the event occurred.  
**uptime** = timestamps will display how long the device had been running when the event occurred.

```
R1(config)#service timestamps log datetime
R1(config)#
R1(config)#service sequence-numbers
R1(config)#exit
R1#
000039: *Feb 11 10:32:46: %SYS-5-CONFIG_I: Configured from console by
jeremy on console
```

### Summary

```

R1(config)# logging console severity
R1(config)# logging monitor severity
R1(config)# logging buffered [size] severity
R1(config)# logging server-ip
R1(config)# logging host server-ip
R1(config)# logging trap severity
R1# terminal monitor
R1(config-line)# logging synchronous
R1(config)# service timestamps log [datetime | uptime]
R1(config)# service sequence-numbers

```

### Syslog vs SNMP

- Syslog and SNMP are both used for monitoring and troubleshooting of devices
  - They are complementary, but their functions are different
- Syslog is used for message logging
  - Events that occurred within the system are categorized based on facility/severity and logged
  - Used for system management, analysis, and troubleshooting
  - Messages are sent from the device to the server
    - Servers can't actively pull info from the device (like SNMP Get) or modify variables (SNMP Set)
- SNMP is used to retrieve and organize information about the SNMP managed devices
  - IP addresses, current interface, temperature, CPU usage, etc
  - SNMP servers can use Get to query the clients and Set to modify variables on the clients

## SSH & Telnet

### Things covered

- Console port security
- Layer 2 switch management IP
- Telnet
- SSH (Secure Shell)

## Console Port Security

### Login

- By default, no password needed to access the CLI of a Cisco IOS device via the console port
- Can configure a password on the console line
  - A user will have to enter a password to access the CLI via the console port

The screenshot shows the configuration of a console line and the resulting user interface:

- Configuration:**

```
R1(config)#line console 0
R1(config-line)#password ccna
R1(config-line)#login
R1(config-line)#end
R1#exit
```
- Output:**

```
R1 con0 is now available
Press RETURN to get started.
```
- User Interface:**

```
User Access Verification
Password:
```
- Annotations:**
  - Yellow box around "R1(config-line)#login": "Tell the device to require a user to enter the configured password to access the CLI via the console port."
  - Yellow box around "User Access Verification Password:" input field: "The password isn't displayed as you type it."

- Single console line: only 1 person can connect to the console at a time

### Login Local

- Alternatively, you can configure the console line to require users to login using one of the configured usernames on the device

The screenshot shows the configuration of a console line to require local login and the resulting user interface:

- Configuration:**

```
R1(config)#username jeremy secret ccnp
R1(config)#line console 0
R1(config-line)#login local
R1(config-line)#end
R1#exit
```
- Output:**

```
R1 con0 is now available
Press RETURN to get started.
```
- User Interface:**

```
User Access Verification
Username: jeremy
Password:
```
- Annotations:**
  - Yellow box around "R1(config-line)#login local": "Tell the device to require a user to login using one of the configured usernames on the device."
  - Yellow box around the configuration block in the output:

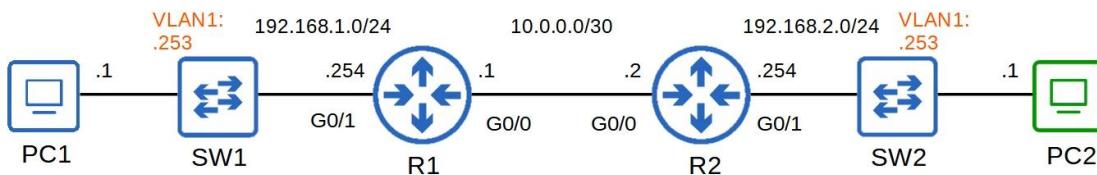
```
line con 0
exec-timeout 3 30
password ccna
logging synchronous
login local
```

with the note: "Log the user out after 3 minutes and 30 seconds of inactivity."

- In the running-config, there is "password ccna", but it won't be used since "login local" activated
  - It will use the password configured for the login local

## Layer 2 Switch - Management IP

- Layer 2 switches don't perform packet routing and don't build a routing table
  - They are not IP routing aware
- However, can assign an IP address to an SVI to allow remote connections to the CLI of the switch
  - E.g. using Telnet or SSH



```
SW1(config)#interface vlan1
SW1(config-if)#ip address 192.168.1.253 255.255.255.0
SW1(config-if)#no shutdown
SW1(config-if)#exit
```

Configure the IP address on the SVI in the same way as on a multilayer switch. Enable the interface if necessary.

```
SW1(config)#ip default-gateway 192.168.1.254
```

Configure the switch's default gateway. In this case, PC2 isn't in the same LAN as SW1. If SW1 doesn't have a default gateway, it can't communicate with PC2.

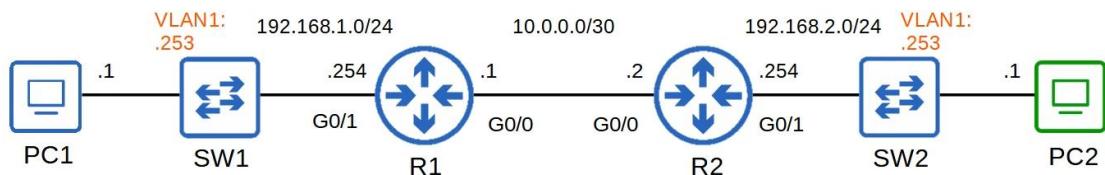
## Telnet (Teletype Network)

- Protocol used to remotely access the CLI of a remote host
- Developed in 1969
- Has been largely replaced by SSH, which is more secure
- Sends data in plain text, no encryption

348 09:38:22.133251 10.0.0.1	10.0.0.2	TELNET	66 Telnet Data ...
> Frame 348: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface -, id 0			
> Ethernet II, Src: 0c:54:cc:2a:0d:00 (0c:54:cc:2a:0d:00), Dst: 0c:54:cc:62:0c:00 (0c:54:cc:62:0c:00)			
> Internet Protocol Version 4, Src: 10.0.0.1, Dst: 10.0.0.2			
> Transmission Control Protocol, Src Port: 23, Dst Port: 28772, Seq: 681, Ack: 33, Len: 12			
▼ Telnet			
Data: \r\n			
Data: Password:			
350 09:38:23.416474 10.0.0.2	10.0.0.1	TELNET	60 Telnet Data ...
> Frame 350: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface -, id 0			
> Ethernet II, Src: 0c:54:cc:62:0c:00 (0c:54:cc:62:0c:00), Dst: 0c:54:cc:2a:0d:00 (0c:54:cc:2a:0d:00)			
> Internet Protocol Version 4, Src: 10.0.0.2, Dst: 10.0.0.1			
> Transmission Control Protocol, Src Port: 28772, Dst Port: 23, Seq: 33, Ack: 693, Len: 4			
▼ Telnet			
Data: ccnp			

- Telnet server (the device being connected to) listens for Telnet traffic on TCP port 23

## Telnet config



```

SW1(config)#enable secret ccna
SW1(config)#username jeremy secret ccna
SW1(config)#access-list 1 permit host 192.168.2.1
SW1(config)#line vty 0 15
SW1(config-line)#login local
SW1(config-line)#exec-timeout 5 0
SW1(config-line)#transport input telnet
SW1(config-line)#access-class 1 in
  
```

If an enable password/secret isn't configured, you won't be able to access privileged exec mode when connecting via Telnet.

Configure an ACL to limit which devices can connect to the VTY lines.

Telnet/SSH access is configured on the VTY lines. There are 16 lines available, so up to 16 users can be connected at once. (VTY stands for Virtual Teletype)

**transport input telnet** allows only Telnet connections.  
**transport input ssh** allows only SSH connections.  
**transport input telnet ssh** allows both.  
**transport input all** allows all connections.  
**transport input none** allows no connections.

Apply the ACL to the VTY lines.  
`*access-class` applies an ACL to the VTY lines,  
`ip access-group` applies an ACL to an interface.

- "line vty 0 15"
  - Means all 16 lines can be used
- SW1(config-line)# access-class 1 in
  - The ACL is only applied for the VTY line
  - Other network devices can still be connected to the switch and won't be blocked as that have to modified on the interface ACL

```

R2#ping 192.168.1.253
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.253, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 10/11/16 ms

R2#telnet 192.168.1.253
Trying 192.168.1.253 ...
% Connection refused by remote host

C:\Users\user>telnet 192.168.1.253
Connecting To 192.168.0.1...
User Access Verification

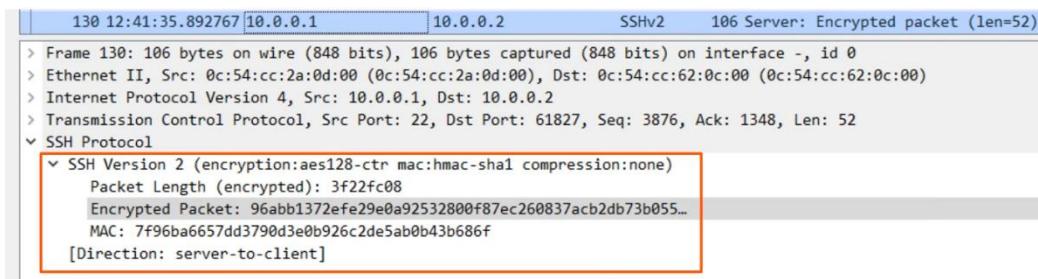
Username: jeremy
Password: 
SW1>
  
```

line vty 0 4  
access-class 1 in  
exec-timeout 5 0  
login local  
transport input telnet  
line vty 5 15  
access-class 1 in  
exec-timeout 5 0  
login local  
transport input telnet

- Older versions only had 5 VTY lines, so that is why they are separated in the running-config

## SSH (Secure Shell)

- Developed in 1995 to replace less secure protocols like Telnet
- Shell
  - A computer program which exposes an operating system's services to a human or other program
- SSHv2, a major revision of SSHv1, released in 2006
- If a device supports both versions, said to run 'version 1.99'
- Provides security features such as data encryption and authentication



- The SSH server (the device being connected to) listens for SSH traffic on port 22

## SSH Config

### Check SSH Support

```
SW1#show version
Cisco IOS Software, vios_12 Software (vios_12-ADVENTERPRISEK9-M), Version 15.2(4.0.55)E, TEST
ENGINEERING ESTG_WEEKLY BUILD, synced to END_OF_FLO_ISP
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2015 by Cisco Systems, Inc.
Compiled Tue 28-Jul-15 18:52 by sasyamal

SW1#show ip ssh
SSH Disabled - version 1.99
%Please create RSA keys to enable SSH (and of atleast 768 bits for SSH v2).
Authentication methods:publickey,keyboard-interactive,password
Authentication Publickey Algorithms:x509v3-ssh-rsa,ssh-rsa
Hostkey Algorithms:x509v3-ssh-rsa,ssh-rsa
Encryption Algorithms:aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc,aes192-cbc,aes256-cbc
MAC Algorithms:hmac-sha1,hmac-sha1-96
Authentication timeout: 120 secs; Authentication retries: 3
Minimum expected Diffie Hellman key size : 1024 bits
IOS Keys in SECSH format(ssh-rsa, base64 encoded): NONE
```

- IOS images that support SSH will have 'K9' in their name
- Cisco exports NPE (No Payload Encryption) IOS images to countries that have restrictions on encryption technologies
- NPE IOS images do not support cryptographies features such as SSH

## RSA Keys

- To enable and use SSH, you must generate an RSA public and private key pair
- The keys are used for data encryption/decryption, authentication, etc

```
SW1(config)#ip domain name jeremysitlab.com
The FQDN of the device is used to name the RSA keys.
FQDN = Fully Qualified Domain Name (host name + domain name)

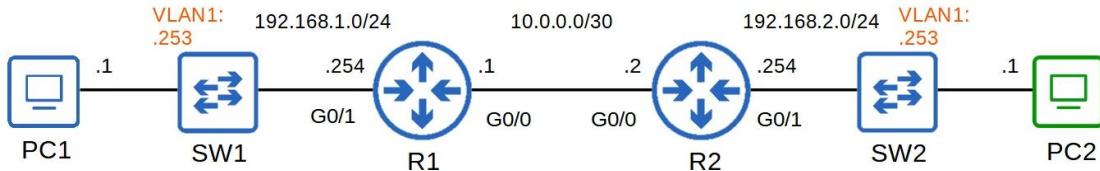
SW1(config)#crypto key generate rsa
The name for the keys will be: SW1.jeremysitlab.com
Choose the size of the key modulus in the range of 360 to 4096 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take
a few minutes.

How many bits in the modulus [512]: 2048
% Generating 2048 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 1 seconds)

SW1(config)#
*Feb 21 04:22:35.778: %SSH-5-ENABLED: SSH 1.99 has been enabled

SW1(config)#do show ip ssh
SSH Enabled - version 1.99
Authentication methods:publickey,keyboard-interactive,password
Encryption Algorithms:aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc,aes192-cbc,aes256-cbc
MAC Algorithms:hmac-sha1,hmac-sha1-96
Authentication timeout: 120 secs; Authentication retries: 3
Minimum expected Diffie Hellman key size : 1024 bits
IOS Keys in SECSh format(ssh-rsa, base64 encoded): SW1.jeremysitlab.com
[output omitted]
```

## VTY Lines



```
SW1(config)#enable secret ccna
SW1(config)#username jeremy secret ccna
SW1(config)#access-list 1 permit host 192.168.2.1

SW1(config)#ip ssh version 2
(optional, but recommended) Restrict SSH to version 2 only.

SW1(config)#line vty 0 15
Configure all VTY lines, just like Telnet.

SW1(config-line)#login local
Enable local user authentication.
*you cannot use login for SSH, only login local.

SW1(config-line)#exec-timeout 5 0
(optional, but recommended) Configure the exec timeout.

SW1(config-line)#transport input ssh
Best practice is to limit VTY line connections to SSH only.

SW1(config-line)#access-class 1 in
(optional, but recommended) Apply the ACL to restrict VTY line connections.
```

## SSH Config Steps

1. Configure host name
2. Configure a DNS domain name
3. Generate RSA key pair
4. Configure enable PW, username/PW
5. Enable SSHv2 (only)
6. Configure VTY lines

- To connect from PC
  - "ssh -l *username ip-address*" OR
  - "ssh *username@ip-address*"

## Summary

```

SW1# show version

SW1# show ip ssh

SW1(config)# ip default-gateway ip-address

SW1(config)# line con 0

SW1(config)# line vty 0 15

SW1(config)# crypto key generate rsa

SW1(config)# ip ssh version 2

SW1(config-line)# login [local]

SW1(config-line)# transport input [protocols | all | none]

SW1(config-line)# exec-timeout minutes seconds

SW1(config-line)# access-class acl in

> telnet ip-address

> ssh -l username ip-address

> ssh username@ip-address

```

## FTP/TFTP

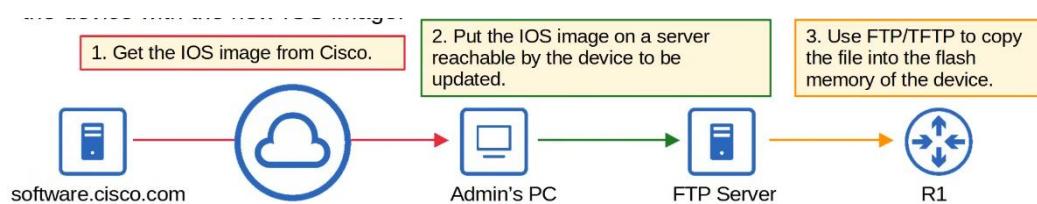
### Things covered

- Purpose of FTP/TFTP

- FTP/TFTP functions and differences
- IOS file system
- Using FTP/TFTP in IOS

## FTP & TFTP

- File Transfer Protocol (FTP)
- Trivial FTP (TFTP)
- Industry standard protocols used to transfer files over the network
- They both use a client-server model
  - Clients can use FTP/TFTP to copy files to/from a server
- As a network engineer, the most common use for FTP/TFTP is in the process of upgrading the OS of a network device
- Can use FTP/TFTP to download newer version of IOS from a server, and then reboot the device with the new IOS image

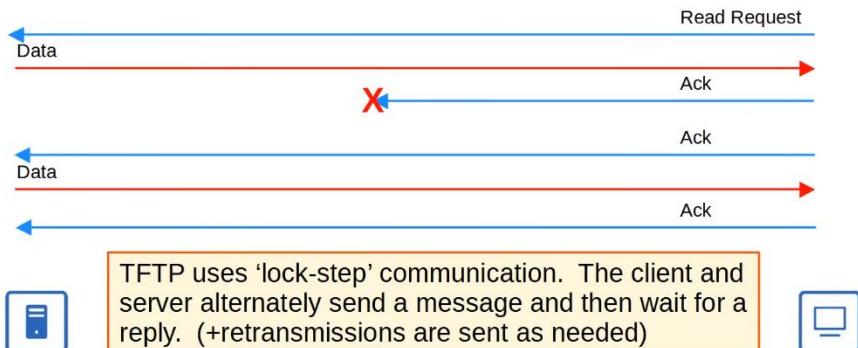


## TFTP

- First standardized in 1981
- Named 'trivial' because it is simple and has only basic features compared to FTP
  - Only allows a client to copy a file to or from a server
- Was released after FTP, but not a replacement for FTP
  - It is another tool to use when lightweight simplicity is more important than functionality
- No authentication (username/PW), so servers will respond to all TFTP requests
- No encryption, so all data is sent in plain text
- Best used in a controlled environment to transfer small files quickly
- TFTP servers listen on UDP port 69
- UDP is connectionless and doesn't provide reliability with retransmissions
- However, TFTP has similar built-in features within the protocol itself

## TFTP Reliability

- Every TFTP data message is acknowledged
  - If client is transferring a file to server, the server will send ACK messages
  - If server is transferring a file to client, the client will send ACK messages
- Timers are used, and if an expected message isn't received in time, the waiting device will re-send its previous message



- Will never send more than 1 message in a row except in retransmission

#### TFTP 'Connections'

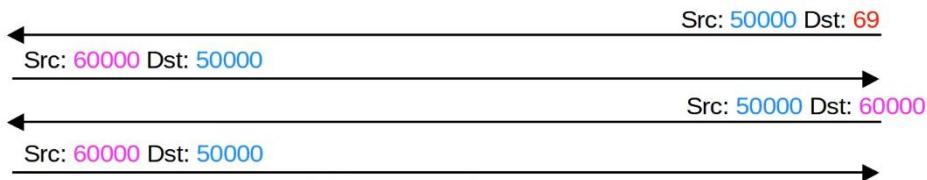
- TFTP file transfers have 3 phases
  1. Connection
    - TFTP client sends a request to the server, and the server responds back, initializing the connection
  2. Data transfer
    - The client and server exchange TFTP messages
    - One sends data and the other sends acknowledgements
  3. Connection Termination
    - After the last data message has been sent, a final acknowledgement is sent to terminate the connection



#### TFTP TID (not in CCNA)

- When the client sends the first message to the server, the destination port is UDP 69 and the source port is a random ephemeral port

- This random port is called a 'Transfer Identifier' (TID) and it identifies the data transfer
- The server then also selects a random TID to use as the source port when it replies, not 69
- When the client sends the next message, the destination port will be the server's TID, not 69

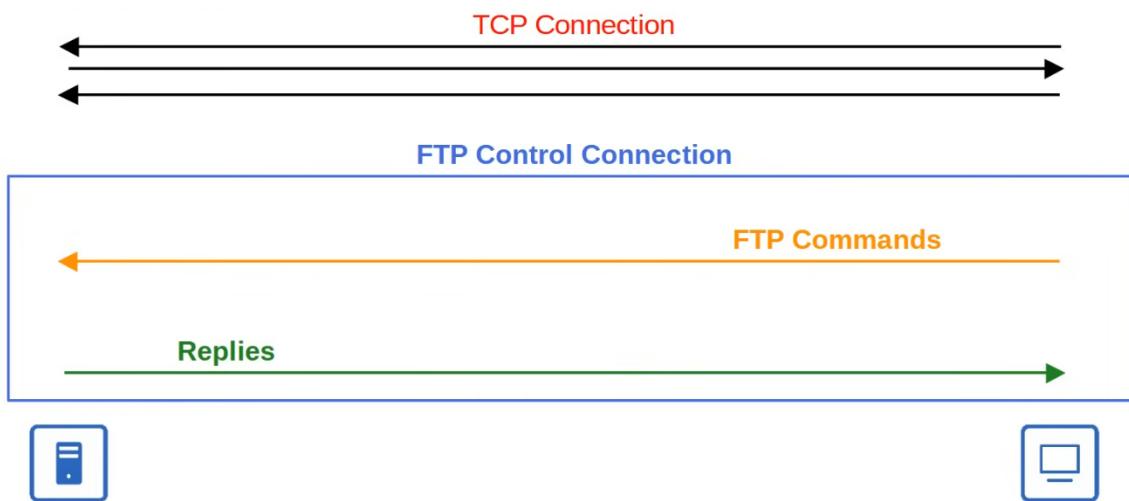


## FTP

- First standardized in 1971
- Uses TCP ports 20 and 21
- Usernames and passwords are used for authentication, however, no encryption
- For greater security, FTPS (FTP over SSL/TLS) can be used
  - Upgrade to FTP
- SSH File Transfer Protocol (SFTP) can also be used for greater security
  - New protocol
- FTP is more complex than TFTP and allows not only file transfers, but clients can also navigate file directories, add and remove directories, list files, etc
- The client sends FTP commands to the server to perform these functions

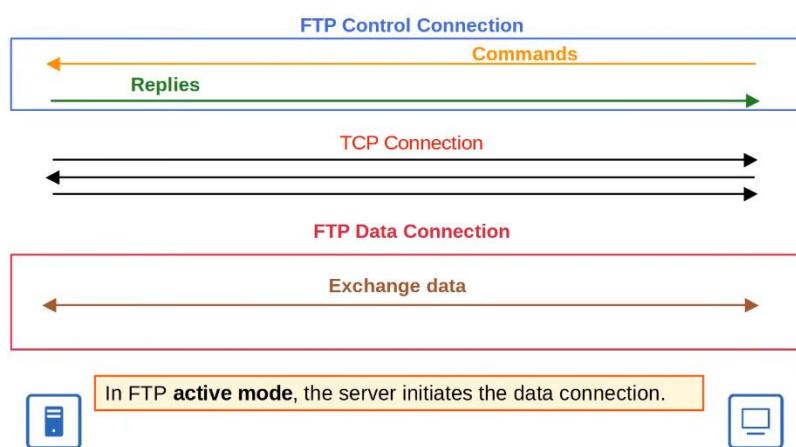
### FTP Control Connections

- FTP uses 2 types of connections
  - An **FTP control connection** (TCP 21) is established and used to send FTP commands and replies
  - When files or data are to be transferred, separate **FTP data connections** (TCP 20) are established and terminated as needed



#### Active Mode FTP Data Connections

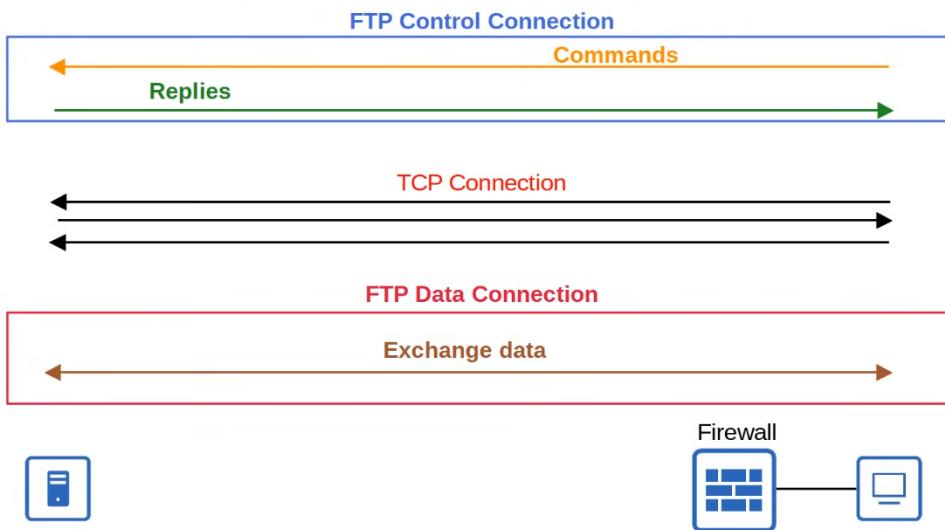
- The default method of establishing FTP data connections is active mode, in which the server initiates the TCP connection



- The Control Connection is not terminated and active throughout, so there are 2 active connections

#### Passive Mode FTP Data Connections

- In FTP passive mode, the client initiates the data connection
- Often necessary when the client is behind a firewall, which could block the incoming connection from the server
- Firewalls usually don't allow 'outside' devices to initiate connections



## FTP vs TFTP

### FTP

- Uses TCP (20 for data, 21 for control) for connection-based communication
- Clients can use FTP commands to perform various actions, not just copy files
- Username/PW authentication
- More complex

### TFTP

- Uses UDP (69) for connectionless communication (although a basic form of 'connection' is used within the protocol itself)
- Clients can only copy files to or from the server
- No authentication
- Simpler

## IOS File Systems

- A file system is a way of controlling how data is stored and retrieved
- Can view the file system of a Cisco IOS device with "show file systems"

```
Router#show file systems
```

#### File Systems:

	Size(b)	Free(b)	Type	Flags	Prefixes
*	2142715904	1994403840	disk	rw	flash0: flash:#
	-	-	disk	rw	flash1:
	966656	962560	disk	rw	flash2:#
	-	-	disk	rw	flash3:
	-	-	opaque	rw	archive:
	-	-	opaque	rw	system:
	262144	256791	nvram	rw	nvram:
	-	-	opaque	rw	tmpsys:
	-	-	network	rw	snmp:
	-	-	opaque	rw	null:
	-	-	network	rw	tftp:
	-	-	opaque	ro	xmodem:
	-	-	opaque	ro	ymodem:
	-	-	opaque	wo	syslog:
	-	-	network	rw	rcp:
	-	-	network	rw	pram:
	-	-	network	rw	ftp:

[output omitted]

o

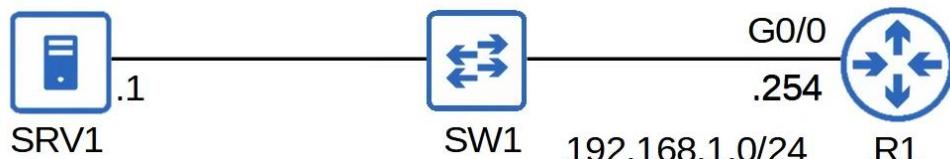
#### Upgrading Cisco IOS

- Can view the current version of IOS with "show version"

```
R1#show version
Cisco IOS Software, C2900 Software [C2900-UNIVERSALK9-M], Version 15.1(4)M4, RELEASE SOFTWARE
(f2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Thurs 5-Jan-12 15:41 by pt_team
[output omitted]
```

- Can view the contents of flash with "show flash"

```
R1#show flash
System flash directory:
File  Length  Name/status
 3  33591768 c2900-universalk9-mz.SPA.151-4.M4.bin
 2  28282    sigdef-category.xml
 1  227537   sigdef-default.xml
[33847587 bytes used, 221896413 available, 255744000 total]
249856K bytes of processor board System flash (Read/Write)
```



```
R1#copy tftp: flash:  
Address or name of remote host []? 192.168.1.1  
Source filename []? c2900-universalk9-mz.SPA.155-3.M4a.bin  
Destination filename [c2900-universalk9-mz.SPA.155-3.M4a.bin]?  
  
Accessing tftp://192.168.1.1/c2900-universalk9-mz.SPA.155-3.M4a.bin....  
Loading c2900-universalk9-mz.SPA.155-3.M4a.bin from  
192.168.1.1: !!!!!!!  
!!!!!!  
!!!!!!  
!!!!!!  
!!!!!!  
[OK - 33591768 bytes]  
  
33591768 bytes copied in 4.01 secs (879550 bytes/sec)
```

copy source destination

Enter the TFTP server IP.

Enter the file name on the server

Enter the name you want to save it as on flash (hit enter to accept the default)

```
R1#show flash  
  
System flash directory:  
File Length Name/status  
3 33591768 c2900-universalk9-mz.SPA.151-4.M4.bin  
4 33591768 c2900-universalk9-mz.SPA.155-3.M4a.bin  
2 28282 sigdef-category.xml  
1 227537 sigdef-default.xml  
[67439355 bytes used, 188304645 available, 255744000 total]  
249856K bytes of processor board System flash (Read/Write)
```

boot system filepath  
\*If you don't use this command, the router will use the first IOS file it finds in flash

```
R1#conf t  
Enter configuration commands, one per line. End with CNTL/Z.  
R1(config)#boot system flash:c2900-universalk9-mz.SPA.155-3.M4a.bin  
R1(config)#exit  
R1#write memory  
Building configuration...  
[OK]  
R1#reload  
Proceed with reload? [confirm]
```

```

R1#show version
Cisco IOS Software, C2900 Software (C2900-UNIVERSALK9-M), Version 15.5(3)M4a, RELEASE SOFTWARE(fc1)
[output omitted]

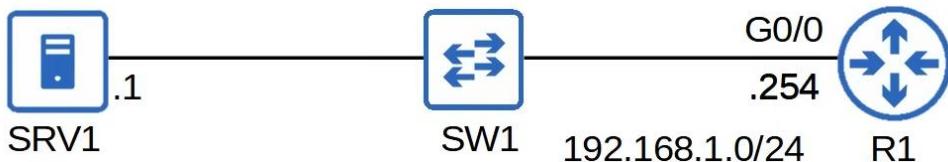
R1#delete flash:c2900-universalk9-mz.SPA.151-4.M4.bin
Delete filename [c2900-universalk9-mz.SPA.151-4.M4.bin]?
Delete flash:/c2900-universalk9-mz.SPA.151-4.M4.bin? [confirm]

R1#show flash

System flash directory:
File  Length  Name/status
 4    33591768 c2900-universalk9-mz.SPA.155-3.M4a.bin
 2    28282    sigdef-category.xml
 1    227537   sigdef-default.xml
[33847587 bytes used, 221896413 available, 255744000 total]
249856K bytes of processor board System flash (Read/Write)

```

### Copying Files (FTP)



```

R1(config)#ip ftp username cisco
R1(config)#ip ftp password cisco
R1(config)#exit

R1#copy ftp: flash:
Address or name of remote host []? 192.168.1.1
Source filename []? c2900-universalk9-mz.SPA.155-3.M4a.bin
Destination filename [c2900-universalk9-mz.SPA.155-3.M4a.bin]?

Accessing ftp://192.168.1.1/c2900-universalk9-mz.SPA.155-3.M4a.bin...
Loading c2900-universalk9-mz.SPA.155-3.M4a.bin from
192.168.1.1: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!![output omitted]

```

Configure the FTP username/password that the device will use when connecting to an FTP server.

### Summary

```
R1# show file systems
R1# show version
R1# show flash
R1# copy source destination
R1(config)# boot system filepath
R1(config)# ip ftp username username
R1(config)# ip ftp password password
```

## Network Address Translation (NAT)

### Static NAT

Things covered

- Private IPv4 addresses
- Intro to NAT
- Static NAT
- Static Nat config

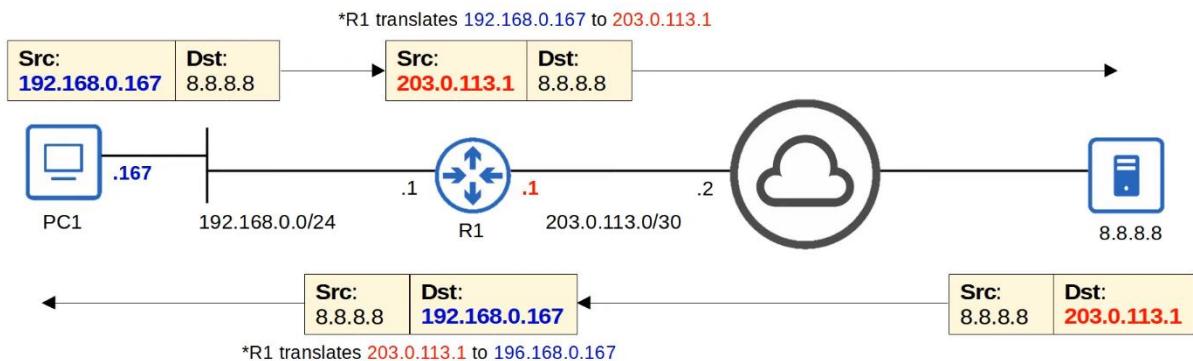
### Private IPv4 Addresses

- IPv4 doesn't provide enough addresses for all devices that need an IP address in the modern world
- Long term solution is to switch to IPv6
- There are 3 main short term solutions
  - CIDR
  - Private IPv4 addresses
  - NAT
- RFC 1918 specifies the following IPv4 address range as private:
  - 10.0.0.0/8 (10.0.0.0 - 10.255.255.255) -> Class A
  - 172.16.0.0/12 (172.16.0.0 - 172.31.255.255) -> Class B
  - 192.168.0.0/16 (192.168.0.0 - 192.168.255.255) -> Class C
- You are free to use these addresses in your networks

- They don't have to be globally unique
- However, they cannot be used over the Internet

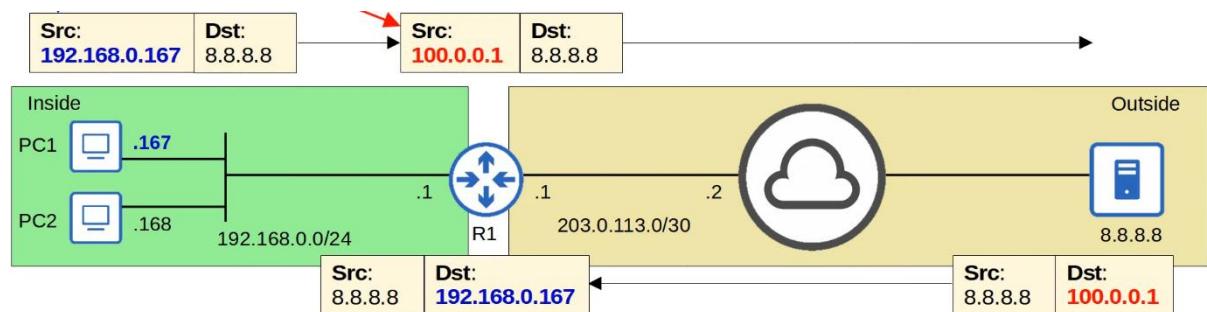
## NAT (Network Address Translation)

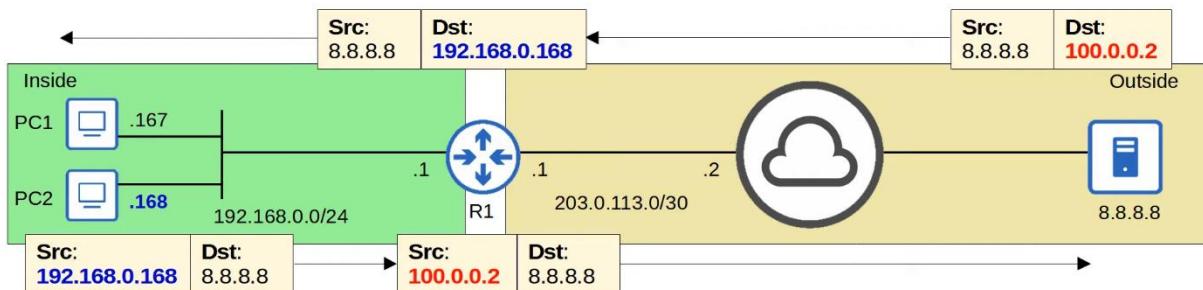
- Used to modify the source and/or destination IP addresses of packets
- There are various reasons to use NAT, but the most common reason is to allow hosts with private IP addresses to communicate with other hosts over the Internet
- Many types of NATs, but for CCNA, need to understand source NAT and how to configure it on Cisco routers



## Static NAT

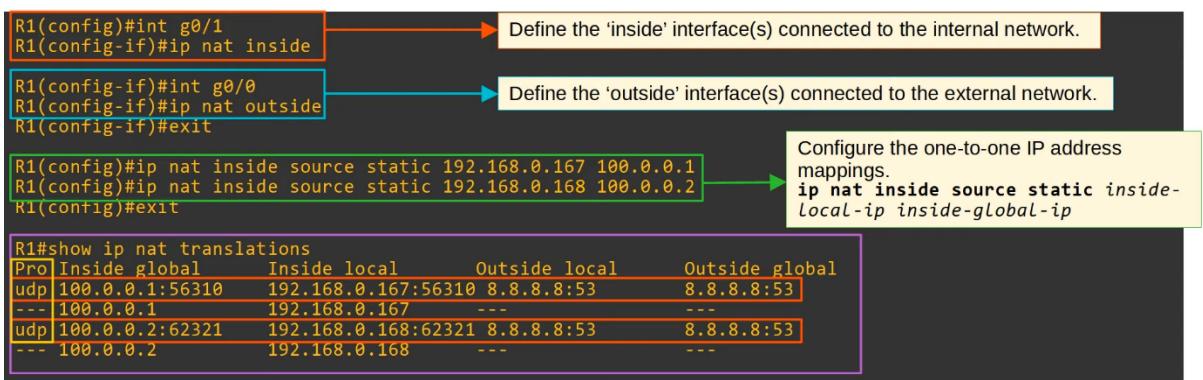
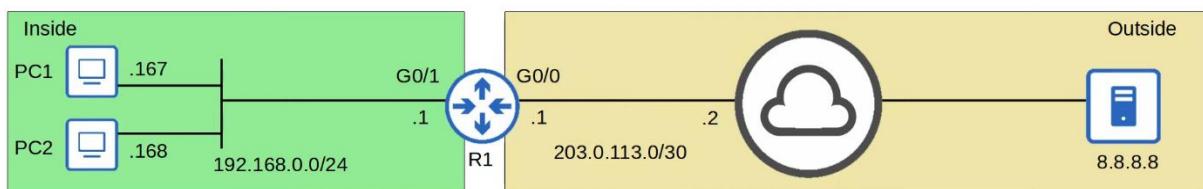
- Involves statically configuring 1-to-1 mappings of private IP addresses to public IP addresses
- An inside local IP address is mapped to an inside global IP address
  - Inside local
    - The IP address of the inside host, from the perspective of the local network
    - The IP address actually configured on the inside host, usually a private address
  - Inside global
    - The IP address of the inside host, from the perspective of the outside network
    - The IP address of the inside host after NAT, usually a public address





- PC1 and PC2 will have different public address
- Static NAT allows devices with private IP addresses to communicate over the Internet, however, because it requires 1-to-1 mapping, it doesn't help preserve IP addresses

### Static NAT Config



- In the "show ip nat translations"
  - There are the static entries and dynamic entries that occur when there is message sent or received
- In the IP address, the additional number is the port number
- Outside local
  - The IP address of the outside host, from the perspective of the local network
- Outside global
  - The IP address of the outside host, from the perspective of the outside network

- Unless 'destination NAT' is used, these 2 addresses will be the same
- Inside/Outside = Location of host
- Local/Global = Perspective

**"clear ip nat translation \*"**

```
R1#show ip nat translations
Pro Inside global      Inside local      Outside local      Outside global
udp 100.0.0.1:56310    192.168.0.167:56310 8.8.8.8:53      8.8.8.8:53
--- 100.0.0.1          192.168.0.167      ---                  ---
udp 100.0.0.2:62321    192.168.0.168:62321 8.8.8.8:53      8.8.8.8:53
--- 100.0.0.2          192.168.0.168      ---                  ---

R1#clear ip nat translation *

R1#show ip nat translations
Pro Inside global      Inside local      Outside local      Outside global
--- 100.0.0.1          192.168.0.167      ---                  ---
--- 100.0.0.2          192.168.0.168      ---                  ---
```

**"show ip nat statistics"**

```
R1#show ip nat statistics
Total active translations: 2 (2 static, 0 dynamic; 0 extended)
Peak translations: 4, occurred 02:29:00 ago
Outside interfaces:
  GigabitEthernet0/0
Inside interfaces:
  GigabitEthernet0/1
Hits: 34  Misses: 0
CEF Translated packets: 30, CEF Punted packets: 4
Expired translations: 4
Dynamic mappings:

Total doors: 0
Appl doors: 0
Normal doors: 0
Queued Packets: 0
```

- Peak translations
  - Most number of entries in the NAT translation table

## Review

```

R1(config-if)# ip nat inside
R1(config-if)# ip nat outside
R1(config)# ip nat inside source static inside-local-ip inside-global-ip
R1# show ip nat translations
R1# show ip nat statistics
R1# clear ip nat translation *

```

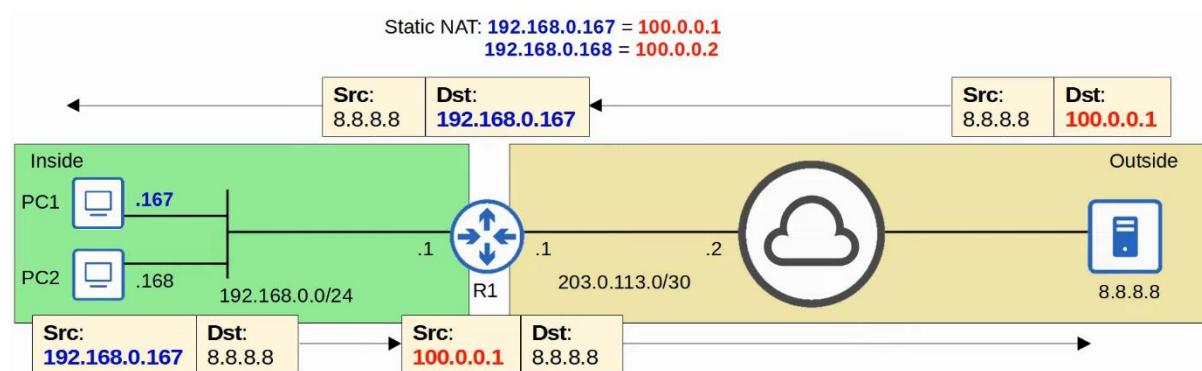
## Dynamic NAT / PAT

Things covered

- More about static NAT
- Dynamic NAT
- Dynamic PAT

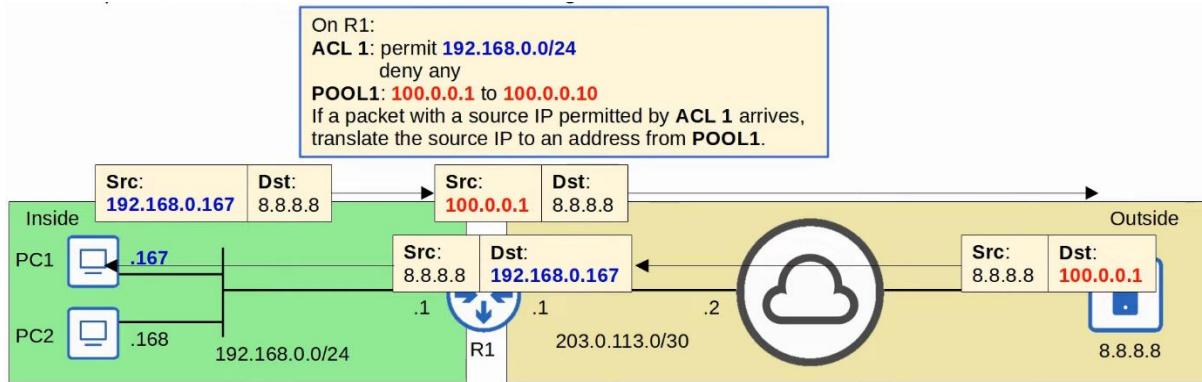
### Static NAT

- Involves statically configuring 1-to-1 mappings of private IP address to public IP addresses
- When traffic from the internal host is sent to the outside network, the router will translate the source address
- This 1-to-1 mapping also allows external hosts to access the internal host via the inside global address



### Dynamic NAT

- The router dynamically maps inside local addresses to inside global addresses as needed
- An ACL is used to identify which traffic should be translated
  - If the source IP is permitted by the ACL, the source IP will be translated
  - If not permitted, source IP will not be translated (traffic will NOT be dropped)
- A NAT pool is used to define the available inside global addresses



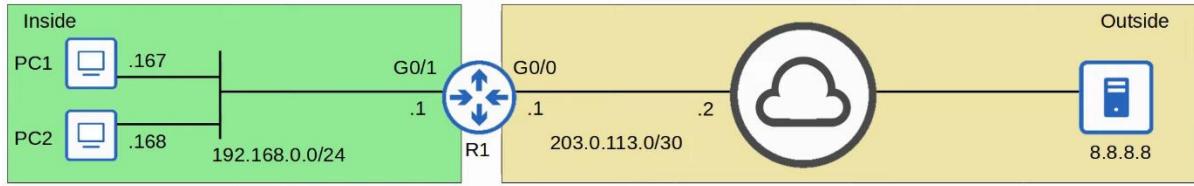
- If there are not enough inside global IP addresses available (all are being used), it is called 'NAT pool exhaustion'
  - If a packet from inside host arrives and need a NAT but there are no available addresses, the router will drop the packet
  - The host will be unable to access outside networks until one of the inside global IP addresses becomes available
  - Dynamic NAT entries will time out automatically if not used, or you can clear them manually

## NAT Pool Exhaustion

Source IP	Translated Source IP
192.168.0.167	100.0.0.1
192.168.0.168	100.0.0.2
192.168.0.100	100.0.0.3
192.168.0.12	100.0.0.4
192.168.0.28	100.0.0.5
192.168.0.56	100.0.0.6
192.168.0.202	100.0.0.7
192.168.0.221	100.0.0.8
192.168.0.116	100.0.0.9
192.168.0.188	100.0.0.10
192.168.0.98	No address available! Router will drop the packet

Source IP	Translated Source IP
192.168.0.168	100.0.0.2
192.168.0.100	100.0.0.3
192.168.0.12	100.0.0.4
192.168.0.28	100.0.0.5
192.168.0.56	100.0.0.6
192.168.0.202	100.0.0.7
192.168.0.221	100.0.0.8
192.168.0.116	100.0.0.9
192.168.0.188	100.0.0.10
192.168.0.98	100.0.0.1

### Dynamic NAT Config



```
R1(config)#int g0/1
R1(config-if)#ip nat inside
Define the 'inside' interface(s) connected to the internal network.

R1(config-if)#int g0/0
R1(config-if)#ip nat outside
R1(config-if)#exit
Define the 'outside' interface(s) connected to the external network.

R1(config)#access-list 1 permit 192.168.0.0 0.0.0.255
Define the traffic that should be translated.
*Traffic permitted by this ACL will be translated.

R1(config)#ip nat pool POOL1 100.0.0.0 100.0.0.255 prefix-length 24
Define the pool of inside global IP addresses.
*Instead of prefix-length 24, you can use netmask 255.255.255.0

R1(config)#ip nat inside source list 1 pool POOL1
Configure dynamic NAT by mapping the ACL to the pool.
```

```
R1#show ip nat translations
Pro Inside global      Inside local        Outside local      Outside global
icmp 100.0.0.1:3       192.168.0.167:3    8.8.8.8:3         8.8.8.8:3
udp 100.0.0.1:58685    192.168.0.167:58685  8.8.8.8:53       8.8.8.8:53
--- 100.0.0.1          192.168.0.167       ---             ---
icmp 100.0.0.2:3       192.168.0.168:3     8.8.8.8:3         8.8.8.8:3
udp 100.0.0.2:49536    192.168.0.168:49536   8.8.8.8:53       8.8.8.8:53
--- 100.0.0.2          192.168.0.168       ---             ---

! below is about 1 minute later

R1#show ip nat translations
Pro Inside global      Inside local        Outside local      Outside global
--- 100.0.0.1           192.168.0.167       ---             ---
--- 100.0.0.2           192.168.0.168       ---             ---
```

- The dynamic addresses will time out after 24 hours
  - Timers can be changed
- The dynamic addresses will also be cleared if command 'clear ip nat translation \*' since they are dynamic and not static

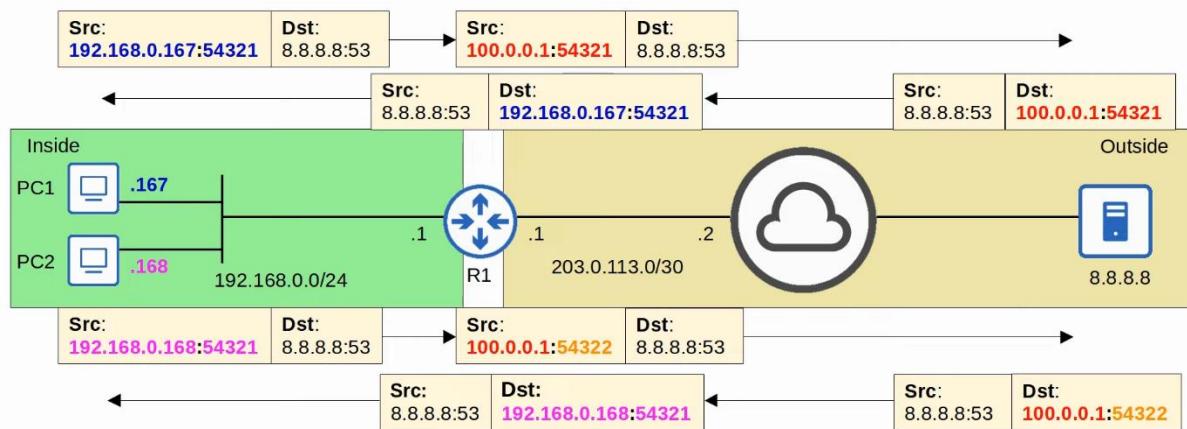
```

R1#show ip nat statistics
Total active translations: 6 (0 static, 6 dynamic; 4 extended)
Peak translations: 6, occurred 00:00:30 ago
Outside interfaces:
  GigabitEthernet0/0
Inside interfaces:
  GigabitEthernet0/1
Hits: 32 Misses: 0
CEF Translated packets: 20, CEF Punted packets: 12
Expired translations: 0
Dynamic mappings:
-- Inside Source
[Id: 1] access-list 1 pool POOL1 refcount 6
pool POOL1: netmask 255.255.255.0
  start 100.0.0.0 end 100.0.0.255
  type generic, total addresses 256, allocated 2 (0%), misses 0
[output omitted]

```

## PAT (Port Address Translation)

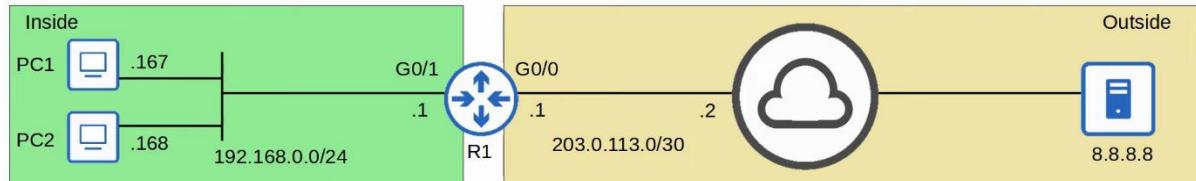
- PAT, also known as NAT overload, translates both the IP address and port number (if necessary)
- By using a unique port number for each communication flow, a single public IP address can be used by many different internal hosts (port number are 16 bits = over 65,000 available port numbers)
- The router will keep track of which inside local address is using which inside global address and port
- Because many inside hosts can share a single public IP, PAT is very useful for preserving public IP addresses, and it is used in networks all over the world



- Router will use a new port number only if the port numbers selected by the end hosts (PCs) are the same, else there is no need

## PAT Config

## Pool



```
R1(config)#int g0/1
R1(config-if)#ip nat inside
```

Define the 'inside' interface(s) connected to the internal network.

```
R1(config-if)#int g0/0
R1(config-if)#ip nat outside
R1(config-if)#exit
```

Define the 'outside' interface(s) connected to the external network.

```
R1(config)#access-list 1 permit 192.168.0.0 0.0.0.255
```

Define the traffic that should be translated.  
\*Traffic permitted by this ACL will be translated.

```
R1(config)#ip nat pool POOL1 100.0.0.0 100.0.0.3 prefix-length 24
```

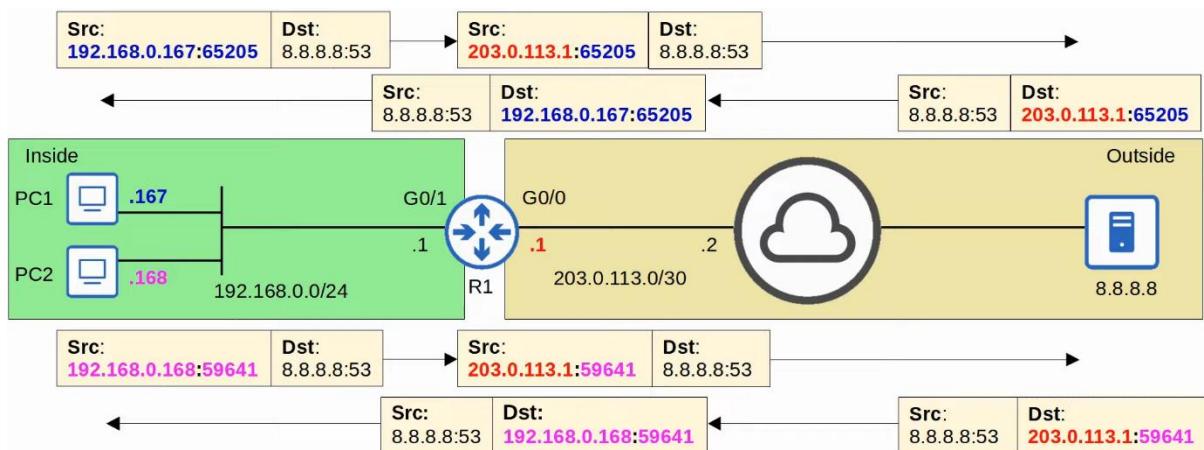
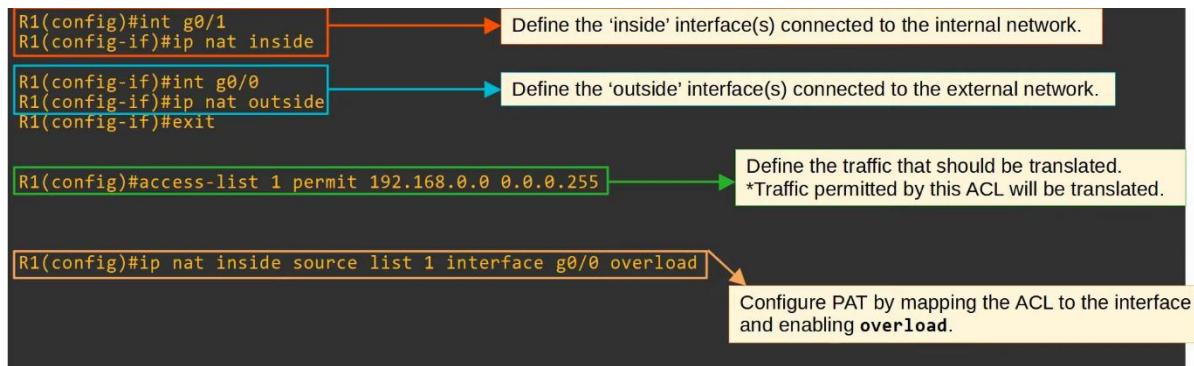
Define the pool of inside global IP addresses.

```
R1(config)#ip nat inside source list 1 pool POOL1 overload
```

Configure PAT by mapping the ACL to the pool and using the **overload** keyword at the end.

```
R1#show ip nat translations
Pro Inside global      Inside local      Outside local      Outside global
  udp 100.0.0.1:63925   192.168.0.167:63925 8.8.8.8:53    8.8.8.8:53
  udp 100.0.0.1:59549   192.168.0.168:59549 8.8.8.8:53    8.8.8.8:53
R1#show ip nat statistics
Total active translations: 2 (0 static, 2 dynamic; 2 extended)
Peak translations: 2, occurred 00:00:03 ago
Outside interfaces:
  GigabitEthernet0/0
Inside interfaces:
  GigabitEthernet0/1
Hits: 4 Misses: 0
CEF Translated packets: 0, CEF Punted packets: 4
Expired translations: 0
Dynamic mappings:
-- Inside Source
[Id: 3] access-list 1 pool POOL1 refcount 2
  pool POOL1: netmask 255.255.255.0
    start 100.0.0.0 end 100.0.0.3
    type generic, total addresses 4, allocated 1 (25%), misses 0
```

## Interface



```

R1#show ip nat translations
Pro Inside global     Inside local      Outside local      Outside global
udp 203.0.113.1:65205 192.168.0.167:65205 8.8.8.8:53      8.8.8.8:53
udp 203.0.113.1:59641 192.168.0.168:59641 8.8.8.8:53      8.8.8.8:53
R1#show ip nat statistics
Total active translations: 2 (0 static, 2 dynamic; 2 extended)
Peak translations: 2, occurred 00:36:30 ago
Outside interfaces:
  GigabitEthernet0/0
Inside interfaces:
  GigabitEthernet0/1
Hits: 12 Misses: 0
CEF Translated packets: 0, CEF Punted packets: 12
Expired translations: 4
Dynamic mappings:
-- Inside Source
[Id: 4] access-list 1 interface GigabitEthernet0/0 refcount 2

```

## Command Review

```

R1(config)# ip nat pool pool-name start-ip end-ip prefix-length prefix-length
R1(config)# ip nat pool pool-name start-ip end-ip netmask subnet-mask
R1(config)# ip nat inside source list access-list pool pool-name
R1(config)# ip nat inside source list access-list pool pool-name overload
R1(config)# ip nat inside source list access-list interface interface overload

```

## Quality of Service (QoS)

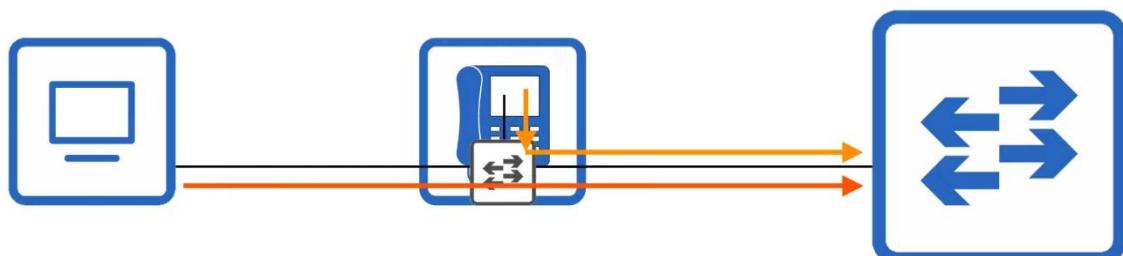
### Intro to QoS

Things covered

- IP Phones/ Voice VLANs
- Power over Ethernet (PoE)
- Intro to Quality of Service (QoS)

### IP Phones

- Traditional phones operate over the public switched telephone network (PSTN)
- Sometimes, this is called POTS (Plain Old Telephone Service)
- IP phones use VoIP (Voice over IP) technologies to enable phone calls over an IP network, such as Internet
- IP phones are connected to a switch just like any other end host
- IP phones have an internal 3-port switch
  - 1 is the 'uplink' to the external switch
  - 1 is the 'downlink' to the PC
  - 1 connects internally to the phone itself
- This allows the PC and IP phone to share a single switch port
  - Traffic from the PC passes through the IP phone to the switch
- It is recommended to separate 'voice' traffic (from the IP phone) and 'data' traffic (from the PC) by placing them in separate VLANs
  - This can be accomplished using a voice VLAN
  - Traffic from the PC will be untagged, but traffic from the phone will be tagged with a VLAN ID



## Voice VLAN

```
SW1(config)#interface gigabitethernet0/0
SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 10
SW1(config-if)#switchport voice vlan 11
```

PC1 will send traffic untagged, as normal.  
SW1 will use CDP to tell PH1 to tag PH1's traffic in VLAN 11.

```
SW1#show interfaces g0/0 switchport
Name: Gi0/0
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 10 (VLAN0010)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: 11 (VLAN0011)
![output omitted]
```

Although the interface sends/receives traffic from two VLANs, it is not considered a trunk port. It is considered an access port.



- SW1 G0/0 is still considered an access port and not a trunk port
- Access port also known as untagged ports

```
SW1#show interfaces trunk
SW1#
SW1#show interfaces g0/0 trunk

Port      Mode          Encapsulation  Status        Native vlan
Gi0/0    off           negotiate      not-trunking  1

Port      Vlans allowed on trunk
Gi0/0    10-11

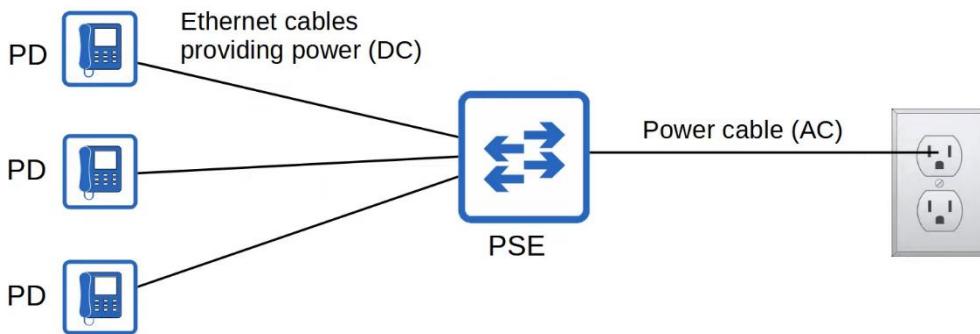
Port      Vlans allowed and active in management domain
Gi0/0    10-11

Port      Vlans in spanning tree forwarding state and not pruned
Gi0/0    10-11
```

## PoE (Power over Ethernet)

- PoE allows Power Sourcing Equipment (PSE) to provide power to Powered Devices (PD) over an Ethernet cable
- Typically, the PSE is a switch and the PDs are IP phones, IP cameras, wireless access points, etc

- The PSE receives AC power from the outlet, converts it to DC power, and supplies that DC power to the PDs
- The same cable is used to provide power and data transfer



- Too much electrical current can damage electrical devices
- PoE has a process to determine if a connected device needs power, and how much power it needs
  - When a device is connected to PoE-enabled port, the PSE sends low power signals, monitors the response, and determine how much power the PD needs
  - If the device needs power, the PSE supplies the power to allow the PD to boot
  - The PSE continues to monitor the PD and supply the required amount of power (but not too much)
- Power policing can be configured to prevent a PD from taking too much power
  - "power inline police"
    - Configures power policing with default settings: disable the port and send a Syslog message if a PD draws too much power
    - Equivalent "power inline police action err-disable"
    - The interface will be put in err-disabled state and can be re-enabled with "shutdown" command followed by "no shutdown"
  - "power inline police action log"
    - Does not shutdown the interface if the PD draws too much power
    - It will restart the interface and send a Syslog message
    - Since restart, PD will lose power and also restart and they will renegotiate power settings

```
SW1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# int g0/0
SW1(config-if)# power inline police
SW1(config-if)# end
SW1# show power inline police g0/0
Available:800(w) Used:32(w) Remaining:768(w)
Interface Admin Oper Admin Oper Cutoff Oper
          State State Police Police Power Power
----- -----
Gi2/1    auto   on    errdisable ok      17.2  16.7
```

```

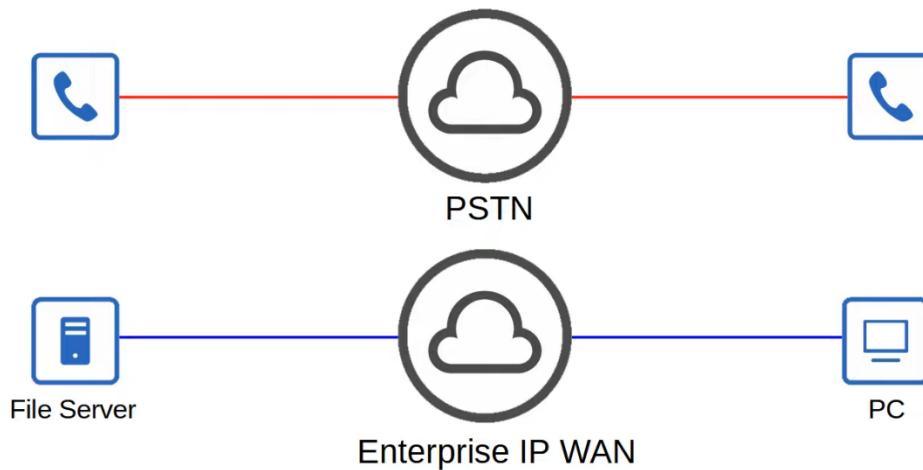
SW1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# int g0/0
SW1(config-if)# power inline police action log
SW1(config-if)# end
SW1# show power inline police g0/0
Available:800(w) Used:32(w) Remaining:768(w)
Interface Admin Oper Admin Oper Cutoff Oper
      State State Police Police Power Power
-----
Gi0/0    auto   on    log      ok     17.2  16.7

```

Name	Standard #	Watts	Powered Wire Pairs
Cisco Inline Power (ILP)	Made by Cisco, not standard	7	2
PoE (Type 1)	802.3af	15	2
PoE+ (Type 2)	802.3at	30	2
UPoE (Type 3)	802.3bt	60	4
UPoE+ (Type 4)	802.3bt	100	4

### QoS (Quality of Service)

- Voice traffic and data traffic used to use entirely separate networks
  - Voice data used the PSTN
  - Data traffic used the IP network (enterprise WAN, Internet, etc)
- QoS wasn't necessary as the different kinds of traffic didn't compete for bandwidth

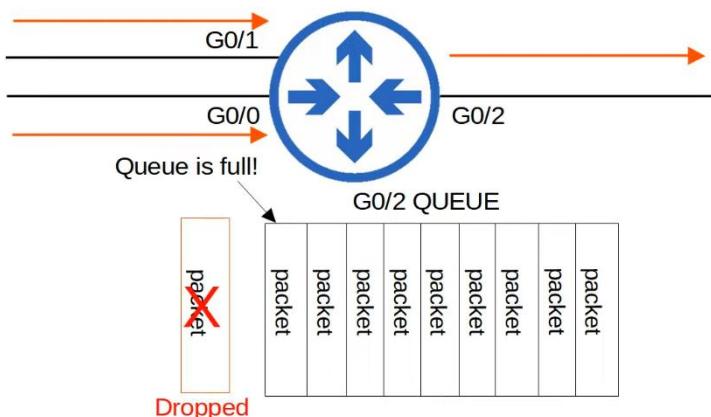


- Modern networks are typically converged networks in which IP phones, video traffic, regular data traffic, etc all share the same IP network
  - This enable cost savings as well as more advanced features for voice and video traffic, for example integrations with collaboration software (Cisco WebEx, Microsoft Teams, etc)
  - However, the different kinds of traffic now have to compete for bandwidth
  - QoS is a set of tools used by network devices to apply different treatment to different packets
- 
- 
- 
- 
- 
- 
- 
- 
- 
- QoS is used to manage the following characteristics of network traffic:
    1. Bandwidth
      - The overall capacity of the link, measured in bps
      - QoS tools allow you to reserve a certain amount of a link's bandwidth for specific kinds of traffic
      - For example, 20% voice traffic, 30% for specific kinds of traffic, and the remaining 50% for all other traffic
    2. Delay
      - The amount of time it takes traffic to go from source to destination: 1-way delay
      - The amount of time it takes traffic to go from source to destination and return: 2-way delay
    3. Jitter
      - The variation in 1-way delay btw packets sent by the same application
      - IP phones have a 'jitter buffer' to provide a fixed delay to audio packets
    4. Loss
      - The % of packets sent that do not reach their destination
      - Can be caused by faulty cables
      - Can also be caused when a device's packet queues get full and the device starts discarding packets
  - The following standards are recommended for acceptable interactive audio (e.g. phone call) quality:

- 1-way delay = 150ms or less
  - Jitter = 30ms or less
  - Loss = 1% or less
- If the standards are not met, there could be a noticeable reduction in the quality of the phone call

## Queueing

- If a network device receives messages faster than it can forward them out of the appropriate interface, the message are placed in a queue
- By default, queued messages will be forwarded in FIFO manner
- If queue is full, new packets are dropped
  - Called 'tail drop'



- Tail drop is harmful because it can lead to TCP global synchronization
- Review of the TCP sliding window
  - Hosts using TCP use the 'sliding window' increase/decrease the rate at which they send traffic as needed
  - When a packet is dropped, it will be re-transmitted
  - When a drop occurs, the sender will reduce the rate it sends traffic
  - It will then gradually increase the rate again
- When the queue fills up and tail drop occurs, all TCP hosts sending traffic will slow down the rate at which they send traffic
- They will all then increase the rate at which they send traffic, which rapidly leads to more congestion, dropped packets, and the process repeats again



- A solution to prevent tail drop and TCP global synchronization is Random Early Detection (RED)

- When the amount of traffic in the queue reaches a certain threshold, the device will start to randomly drop packets from select TCP flows
- Those TCP flows that dropped packets will reduce the rate at which traffic is sent, but you will avoid global TCP synchronization, in which ALL TCP flows reduce and then increase the rate of transmission at the same time in waves
- In standard RED, all kinds of traffic are treated the same
- An improved version, Weighted RED (WRED), allows you to control which packets are dropped depending on the traffic class

## **QoS (Part 2)**

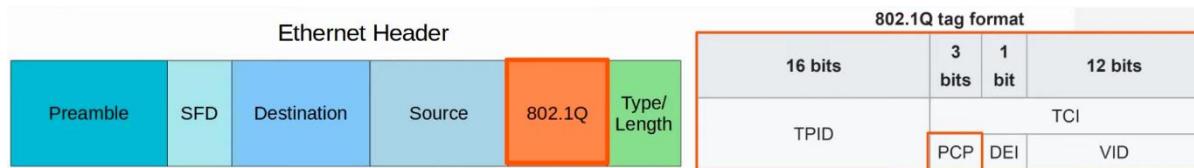
Things covered

- Classification/ Marking
- Queueing / Congestion Management
- Shaping / Policing

### **Classification**

- The purpose of QoS is to give certain kinds of network traffic priority over others during congestion
- Classification organizes network traffic (packets) into traffic classes (categories)
- Classification is fundamental to QoS. To give priority to certain types of traffic, you have to identify which type of traffic to give priority to
- There are many methods of classifying traffic. For example
  - ACL
    - Traffic permitted by the ACL will be given certain treatment, other traffic will not
  - NBAR (Network Based Application Recognition)
    - Performs a deep packet inspection, looking beyond the Layer 3 and 4 information layer up to Layer 7 to identify specific kinds of traffic
    - In the layer 2 and 3 headers, there are specific fields used for this purpose
- The PCP (Priority Code Point) field of the 802.1Q tag (in the Ethernet header) can be used to identify high/low priority traffic
  - Only when there is a dot1q tag
- The DSCP (Differentiated Services Code Point) field of the IP header can also be used to identify high/low priority traffic

### **PCP / CoS**

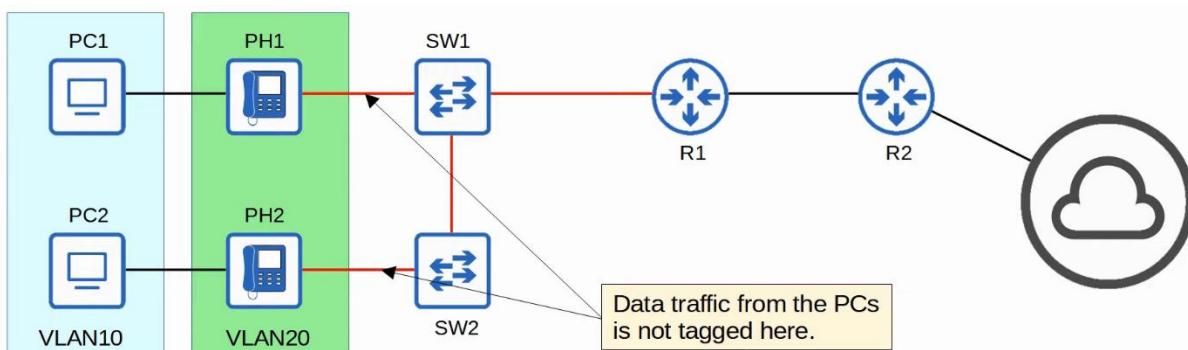


- PCP is also known as Class of Service (CoS)
- Defined by IEEE 802.1p
- 3 bits = 8 possible values

PCP value	Traffic types
0	Best effort (default)
1	Background
2	Excellent effort
3	Critical applications
4	Video
5	Voice
6	Internet control
7	Network control

- 0: Best effort
  - 'Best effort' delivery means there is no guarantee that data is delivered or that it meets any QoS standard
  - This is regular traffic, not high-priority
- IP phones **mark** call signalling traffic (used to establish calls) as PCP 3
- They **mark** the actual voice traffic as PCP 5

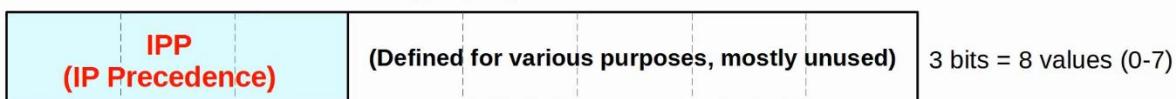
- Because PCP is found in the dot1q header, it can only be used over the following connections:
  - Trunk links
  - Access links with a voice VLAN
- In the diagram below, traffic btw R1 and R2, or btw R2 and external destinations will not have a dot1q tag. So, traffic over those links PCP cannot be marked with a PCP value



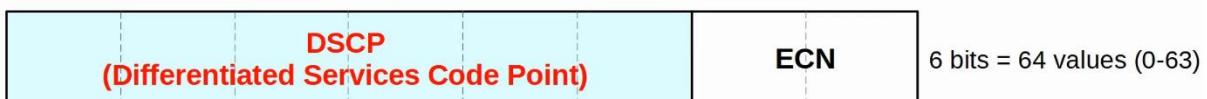
## IP Type of Service (ToS) Byte

Offsets	Octet	0	1	2	3		
Octet	Bit	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23 24	25 26 27 28 29 30 31		
0	0	Version	IHL	DSCP	ECN	Total Length	
4	32		Identification		Flags	Fragment Offset	
8	64	Time To Live		Protocol		Header Checksum	
12	96			Source IP Address			
16	128			Destination IP Address			
20	160						
24	192						
28	224					Options (if IHL > 5)	
32	256						

ToS byte (old)



ToS byte (current)



## IP Precedence

IPP (IP Precedence)	(Defined for various purposes, mostly unused)	3 bits = 8 values (0-7)
------------------------	---	-------------------------

- Standard IPP markings are similar to PCP
  - 6 and 7 are reserved for 'network control' traffic (i.e. OSPF messages btw routers)
  - 5: voice
  - 4: video
  - 3: voice signalling
  - 0: best effort
- With 6 and 7 reserved, 6 possible values remain
- Although 6 values is sufficient for many networks, the QoS requirements of some networks demand more flexibility

## DSCP

DSCP (Differentiated Services Code Point)	ECN	6 bits = 64 values (0-63)
--	-----	---------------------------

- RFC 2474 (1998) defines the DSCP field, and other 'DiffServ' RFCs elaborate on its use
- With IPP updated to DSCP, new standard markings had to be decided upon
  - By having generally agreed upon standard markings for different kinds of traffic, QoS design & implementation is simplified, QoS works better btw ISPs and enterprises, among other benefits
- You should be aware of the following standard markings
  - Default Forwarding (DF) - best effort traffic
  - Expedited Forwarding (EF) - low loss/latency/jitter traffic (usually voice)
  - Assured Forwarding (AF) - A set of 12 standard values
  - Class Selector (CS) - A set of 8 standard values, provides backward compatibility with IPP

```
R1(config)#class-map TEST
R1(config-cmap)#match dscp ?
<0-63> Differentiated services codepoint value
af11 Match packets with AF11 dscp (001010)
af12 Match packets with AF12 dscp (001100)
af13 Match packets with AF13 dscp (001110)
af21 Match packets with AF21 dscp (010010)
af22 Match packets with AF22 dscp (010100)
af23 Match packets with AF23 dscp (010110)
af31 Match packets with AF31 dscp (011010)
af32 Match packets with AF32 dscp (011100)
af33 Match packets with AF33 dscp (011110)
af41 Match packets with AF41 dscp (100010)
af42 Match packets with AF42 dscp (100100)
af43 Match packets with AF43 dscp (100110)
cs1 Match packets with CS1(precedence 1) dscp (001000)
cs2 Match packets with CS2(precedence 2) dscp (010000)
cs3 Match packets with CS3(precedence 3) dscp (011000)
cs4 Match packets with CS4(precedence 4) dscp (100000)
cs5 Match packets with CS5(precedence 5) dscp (101000)
cs6 Match packets with CS6(precedence 6) dscp (110000)
cs7 Match packets with CS7(precedence 7) dscp (111000)
default Match packets with default dscp (000000)
ef Match packets with EF dscp (101110)
```

### DF (Default Forwarding)

32	16	8	4	2	1	
0	0	0	0	0	0	

- DF is used for best-effort traffic
- The DSCP marking for DF is 0

### EF (Expedited Forwarding)

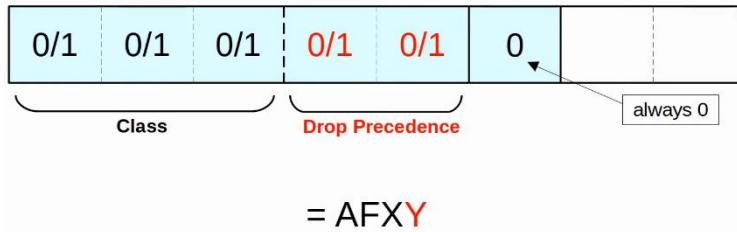
32	16	8	4	2	1	
1	0	1	1	1	0	

- EF is used for traffic that require low loss/latency/jitter

- The DSCP marking for EF is 46

## AF (Assured Forwarding)

- AF defines 4 traffic classes
  - All packets in a class have the same priority
- Within each class, there are 3 levels of drop precedence
  - Higher drop precedence = more likely to drop the packet during congestion
- Max value = AF43, DSCP 38
- Formula: DSCP =  $8X + 2Y$



## Example

(32)	(16)	(8)	(4)	(2)	(1)
4	2	1	2	1	
0	0	1	0	1	0

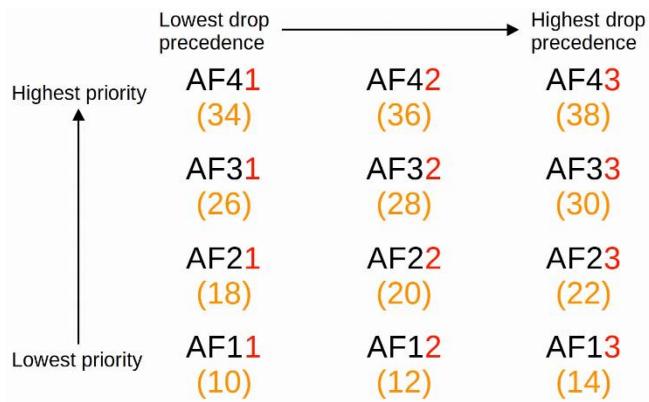
$= \text{AF}11$

(DSCP 10)

(32)	(16)	(8)	(4)	(2)	(1)
4	2	1	2	1	
0	1	0	1	1	0

$= \text{AF}23$

(DSCP 22)



### CS (Class Selector)

- CS defines 8 DSCP values for backward compatibility with IPP
- The 3 bits that were added for DSCP are set to 0, and the original IPP bits are used to make 8 values

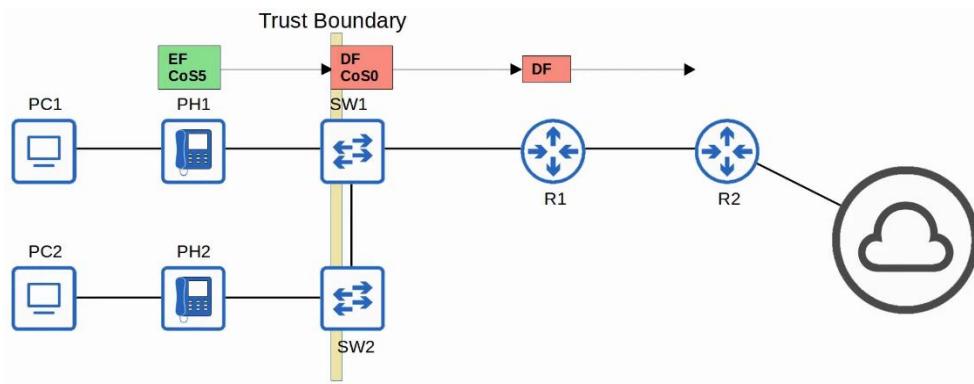
	(32) 4	(16) 2	(8) 1	(4) 0	(2) 0			
IPP:	0	1	2	3	4	5	7	
cs:	CS0	CS1	CS2	CS3	CS4	CS5	CS6	CS7
DSCP: (decimal)	0	8	16	24	32	40	48	56

### RFC 4954

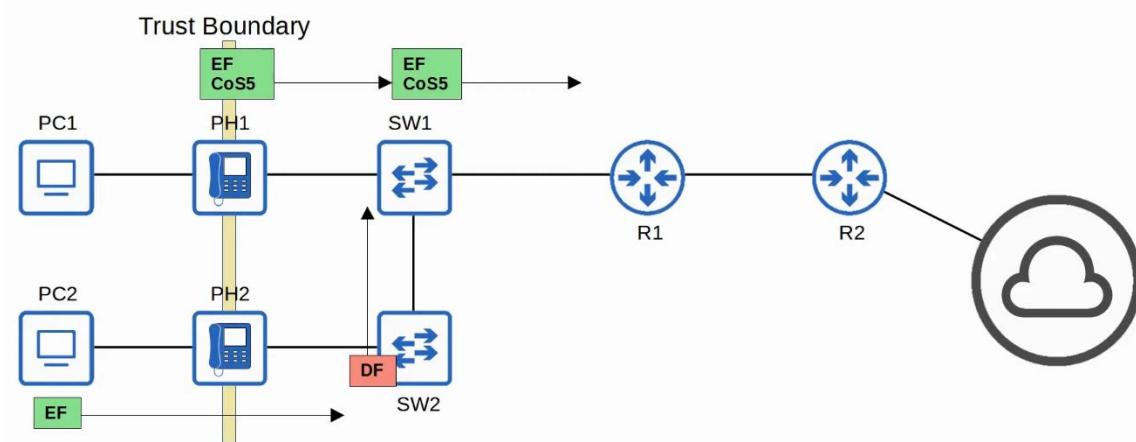
- Was developed with the help of Cisco to bring all of these values together and standardize their use
- The RFC offers many specific recommendations, but here are a few key ones:
  - Voice traffic: EF
  - Interactive video: AF4X
  - Streaming video: AF3X
  - High priority data: AF2X
  - Best effort: DF
  - X: can be any number

## Trust Boundaries

- The trust boundary of a network defines where devices trust/don't trust the QoS markings of received messages
- If the markings are trusted, the device will forward the message without changing the markings
- If the markings aren't trusted, the device will change the markings according to the configured policy

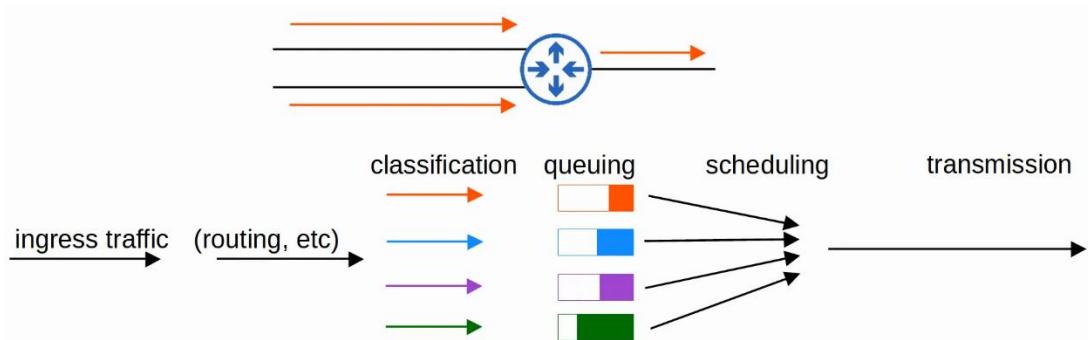


- If an IP phone is connected to the switch port, it is recommended to move the trust boundary to the IP phones
- This is done via configuration on the switch port connected to the IP phone
- If a user marks their PC's traffic with a high priority, the marking will be changed (not trusted)

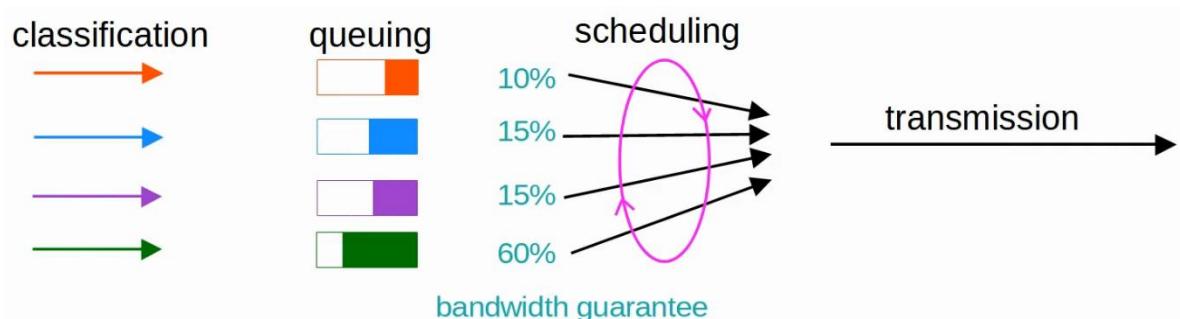


## Queueing / Congestion Management

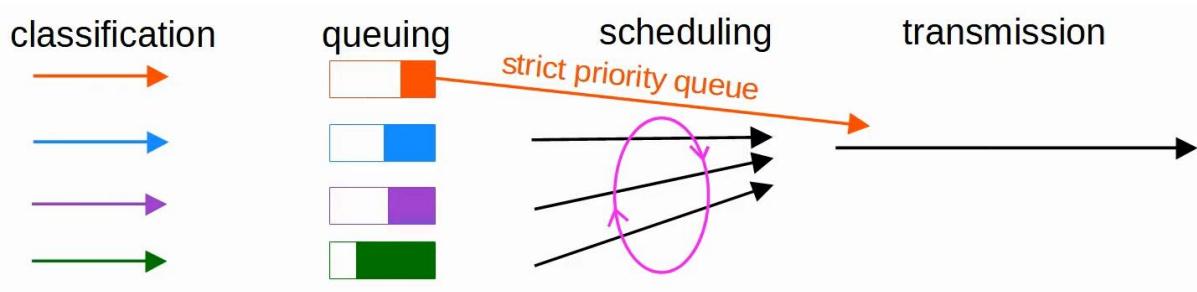
- When a network device receives traffic faster at a faster rate than it can forward the traffic out of the appropriate interface, packets are placed in that interface's queue as they wait to be forwarded
- When the queue becomes full, packets that don't fit in the queue are dropped (tail drop)
- RED and WRED drop packets early to avoid tail drop
- An essential part of QoS is the use of multiple queues
  - This is where classification plays a role
  - The device can match traffic based on various factors (e.g. DSCP marking in IP header) and then place it in the appropriate interface
- However, the device is only able to forward one frame out of an interface at a time, so a scheduler is used to decide which queue traffic is forwarded from next
  - Prioritization allows the scheduler to give certain queues more priority than others



- A common scheduling method is weighted round robin
  - Round-robin = packets are taken from each queue in order, cyclically
  - Weighted = more data is taken from high priority queues each time the scheduler reaches that queue
- CBWFQ (Class-Based Weighted Fair Queueing)
  - Popular method of scheduling
  - Uses a weighted round-robin scheduler while guaranteeing each queue a certain percentage of the interface's bandwidth during congestion

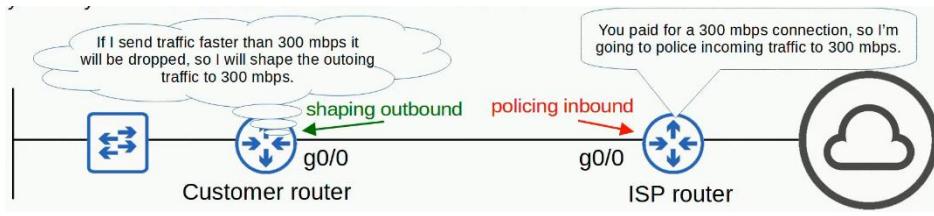


- Round robin scheduling is not ideal for voice/video traffic
  - Even if the voice/video traffic receives a guaranteed minimum amount of bandwidth, round robin can add delay and jitter because even the high priority queues have to wait their turn in the scheduler
- LLQ (Low Latency Queueing) designates 1 (or more) queues as strict priority queues
  - This means that if there is traffic in the queue, the scheduler will ALWAYS take the next packet from that queue until it is empty
- This is very effective for reducing delay and jitter of voice/video traffic
- However, it has the downside of potentially starving other queues if there is always traffic in the designated strict priority queue
  - Policing can control the amount of traffic allowed in the strict priority queue so that it can't take all of the link's bandwidth



## Shaping and Policing

- Traffic shaping and policing are both used to control the rate of traffic
- Shaping buffers traffic in a queue if the traffic rate goes over the configured rate
- Policing drops traffic if the traffic rate goes over the configured rate
  - Note: policing also has the option of re-marking the traffic instead of dropping it
  - 'Burst' traffic over the configured rate is allowed for short period of time
  - This accommodates data applications which typically are 'bursty' in nature. Instead of a constant stream of data, they send data in bursts
  - The amount of burst traffic allowed is configurable
- In both cases, classification can be used to allow for different rates for different kinds of traffic
- Why would you want to limit the rate traffic is sent/received?



## Security Fundamentals

Things covered

- Key security concepts
- Common attacks
- Password / Multi-Factor Authentication (MFA)
- Authentication, Authorization, Accounting (AAA)
- Security Program Elements

### Why security?

- What is the purpose/goal of security in an enterprise?
- The principles of the CIA Triad form the foundation of security
  - Confidentiality
    - Only authorized users should be able to access data
    - Some information/data is public and can be accessed by everyone
    - Some are secret and can only be accessed by certain people
  - Integrity
    - Data should not be tampered/modified by unauthorized users
    - Data should be authentic and correct
  - Availability
    - The network/systems should be operational and accessible to authorized users
- Attackers can threaten the CIA of an enterprise's systems and information
- Vulnerability
  - Any potential weakness that can compromise the CIA of a system/info
  - A potential weakness isn't a problem on its own
- Exploit
  - Something that can potentially be used to exploit the vulnerability
  - An exploit isn't a problem on its own

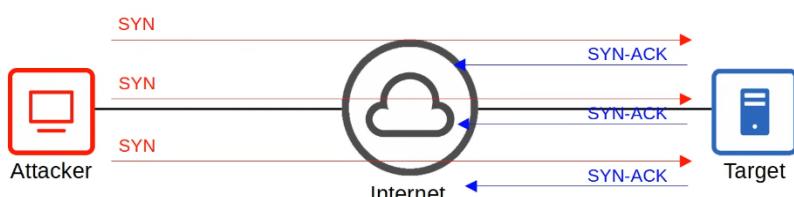
- A threat
  - The potential of a vulnerability to be exploited
  - E.g. A hacker exploiting a vulnerability in your system is a threat
- Mitigation Technique
  - Something that can protect against threats
  - Should be implemented everywhere a vulnerability can be exploited: Client devices, servers, switches, routers, firewalls, etc
- No system is perfectly safe

## Common Attacks

- DoS (denial-of-service)
- Spoofing
- Reflection/amplification
- Man-in-the-middle
- Reconnaissance
- Malware
- Social Engineering
- Password-related

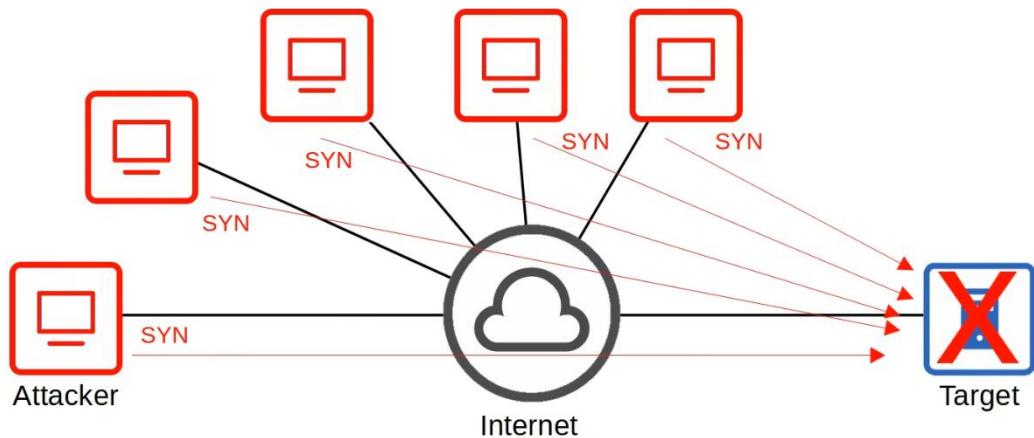
## DoS (Denial-of-Service)

- Threatens the availability of a system
- A common DoS attack is TCP SYN flood
  - TCP 3-way handshake: SYN / SYN-ACK / ACK
  - Attacker send lots of SYN messages but don't acknowledge them, which causes the target run out of TCP connections
  - How it works
    - Attacker sends countless TCP SYN messages to the target
    - Target send SYN-ACK in response to each SYN message received
    - Attacker never replies with the final ACK of the 3-way handshake
    - The incomplete connection fills up the target's TCP connection table
    - Attacker continues to send SYN messages
    - The target is no longer able to make legitimate TCP connections



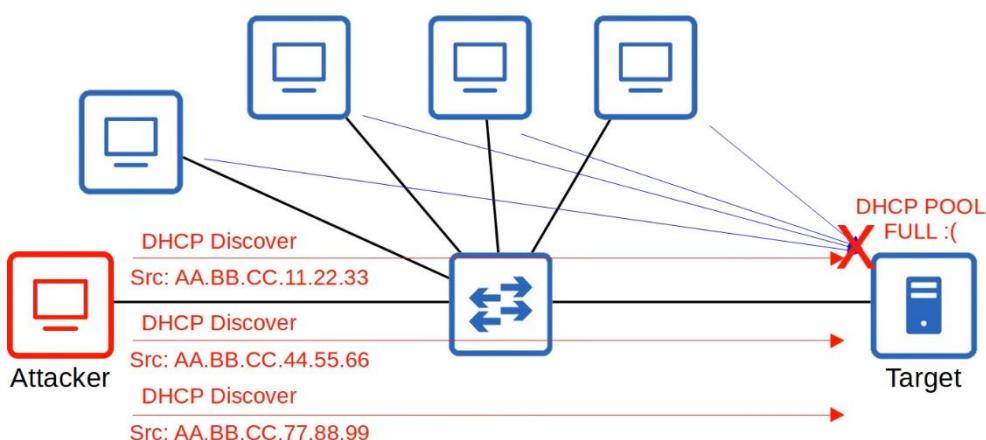
- Note: SYN-ACK not returning to Attacker directly as most likely attacker has spoofed his IP address

- More common is DDoS (Distributed DoS)
  - Attacker infects many target computers with malware and uses them all to initiate a DoS attack
  - The group of infected computers is called botnet



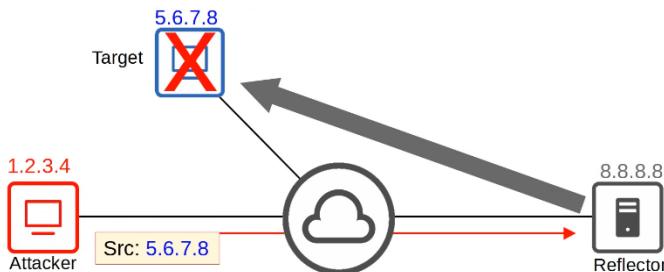
## Spoofing

- To spoof an address is to use a fake source address (IP/MAC address)
- Numerous attacks involve spoofing, it is not a single kind of attack
- Example: DHCP exhaustion
  - An attacker uses spoofed MAC addresses to flood DHCP Discover messages
  - The target server's DHCP pool becomes full, resulting in DoS to other devices



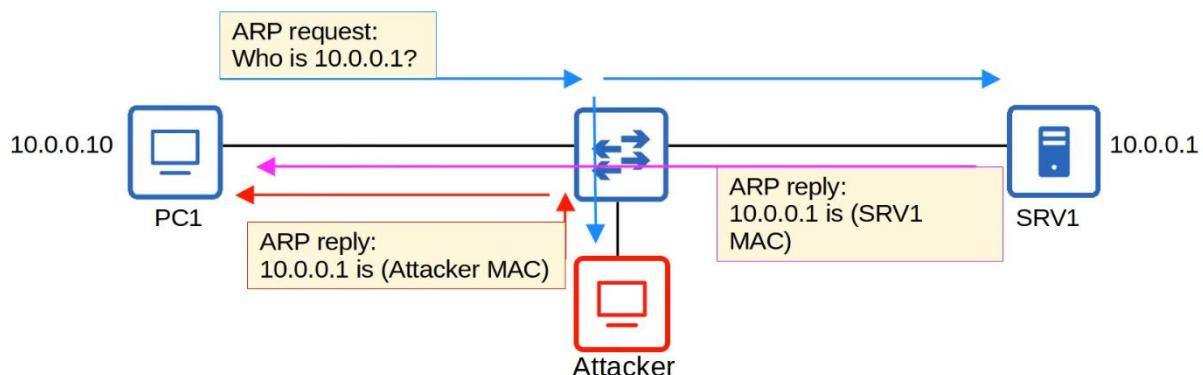
## Reflection/Amplification

- In a reflection attack, the attacker sends traffic to a reflector, and spoofs the source address of its packets using the target's IP address
- The reflector (i.e. DNS server) sends the reply to the target's IP address
- If the amount of traffic sent to the target is large enough, can cause a DoS
- A reflection attack becomes an amplification attack when the amount of traffic sent by the attacker is small, but it triggers a large amount of traffic to be sent from the reflector to the target
  - Example: DNS/NTP amplification attack

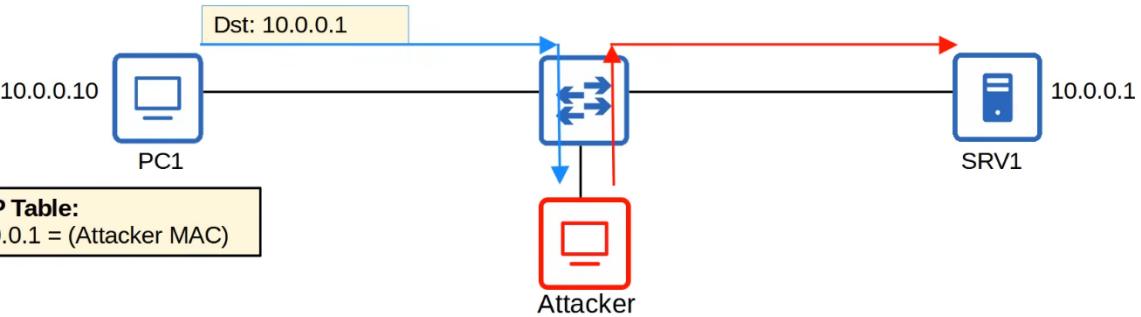


### Man-in-the-middle

- The attacker places himself btw the source and destination to eavesdrop on communications, or to modify traffic before it reaches the destination
- Common example: ARP Spoofing/Poisoning
  - Host sends an ARP Request, asking for MAC address of another device
  - The target of the request sends an ARP Reply, informing the requester of its MAC address
  - The attacker waits and sends another ARP Reply after the legitimate replier
  - If the attacker's ARP reply arrives last, it will overwrite the legitimate ARP entry in PC1's ARP table



- In PC1's ARP table, the entry for 10.0.0.1 will have the attacker's MAC address
- When PC1 tries to send traffic to SRV1, it will be forwarded to the attacker instead
- The attacker can modify/inspect the message before forwarding them to SRV1
- This compromises the Confidentiality and Integrity of communications btw PC1 and SRV1



## Reconnaissance

- Reconnaissance attacks aren't attacks themselves, but they are used to gather information about a target which can be used for a future attack
- This is often publicly available information
- E.g. "nslookup" to learn IP address of a site

```
C:\Users\user>nslookup jeremysitlab.com
Server: UnKnown
Address: 192.168.0.1

Non-authoritative answer:
Name: jeremysitlab.com
Address: 162.241.216.233
```

- E.g. WHOIS query to learn email addresses, phone numbers, physical address, etc

<https://lookup.icann.org/lookup>

Domain Information	
Name:	jeremysitlab.com
Region:	Domain ID:755991540, F0944N, COM-VTSV
Device Status:	<a href="#">Device status</a>
Newspaper:	<a href="#">Newspaper</a>
WHOIS:	<a href="#">WHOIS</a>
MX Record:	<a href="#">MX Record</a>
Dates	
Registry Expiration:	2024-09-20 11:22:29 UTC
Created:	2020-09-09 14:23:29 UTC

Contact Information	
Registrant:	Name: Domain Privacy Service FBO Registrant, Email: whois@domainprivacy.com
Status:	Active
Phone:	+1-800-7057640
Fax:	
Kind:	Individual
Mailing Address:	10 CORPORATE DR, SUITE 200, DUXBURY, Massachusetts, 02332, US

## Malware (Malicious Software)

- Refers to a variety of harmful programs that can infect a computer
- Viruses
  - They infect other software ("host program")
  - The virus spread as the software is shared by users
  - Typically corrupt or modify files on the target computer
- Worms
  - Do not require a host program

- They are a standalone malware and they are able to spread on their own w/o user interaction
  - The spread of worms can congest the network, but the 'payload' of a worm can cause additional harm to target devices
- Trojan Horse
  - Harmful software that is disguised as legitimate software
  - They spread through user interaction such as opening email attachments, or downloading a file from the Internet
- These malware types can exploit various vulnerabilities to threaten any of the CIA of the target device
- There are many other types of malware

## **Social Engineering**

- Attacks the most vulnerable part of any system: people
- They involve psychological manipulation to make the target reveal confidential information or perform some action
- Phishing
  - Typically involves fraudulent emails that appear to come from a legitimate business and contain links to a fraudulent website that seems legitimate
  - Users are told to login to the fraudulent website, providing their login credentials to the attacker
  - Spear phishing
    - A more targeted form of phishing, i.e. aimed at employees of a certain company
  - Vishing (voice phishing)
    - Phishing performed over the phone
    - 'Hi, this is Jeremy from the IT department. Due to company policy we need to reset your password, could you tell me the password you're currently using and I'll reset it for you?'
  - Smishing (SMS phishing)
    - phishing using SMS text messages.
- Watering hole attacks
  - Compromises sites that the target victim frequently visits
  - If a malicious link is placed on a website the target trusts, they might not hesitate to click it
- Tailgating
  - Involve entering restricted, secured areas by simply walking in behind an authorized person as they enter

## **Password-related**

- Most systems use a username/password combination to authenticate users
- The username is often simple/easy to guess (e.g. user's email), and the strength and secrecy of the password is relied on to provide the necessary security

- Attackers can learn a user's password via multiple methods
  - Guessing
  - Dictionary attack
    - Use a dictionary of common words/passwords
  - Brute force attack
    - Try every possible combination of characters
- Strong password should contain
  - $\geq 8$  characters
  - Mix of upper/lower case
  - Mix of letters/numbers
  - $\geq 1$  special character
  - Changed regularly

- DOS (denial-of-service) attacks
  - target the availability of a system so users can't access it
- Spoofing attacks
  - involve using fraudulent source IP/MAC addresses
- Reflection/amplification attacks
  - involve spoofing a source IP address to cause a reflector to send lots of traffic to the target
- Man-in-the-middle attacks
  - an attacker intercepts traffic between the source and destination to eavesdrop and/or modify the traffic
- Reconnaissance attacks
  - used to gather information on the target to perform future attacks
- Malware
  - malicious software such as viruses, worms, and trojan horses that infect a system
- Social engineering attacks
  - attacks that use psychological manipulation to target people and make them reveal info or perform an action
- Password-related attacks
  - attacks such as dictionary attacks and brute force attacks, used to guess the target's password

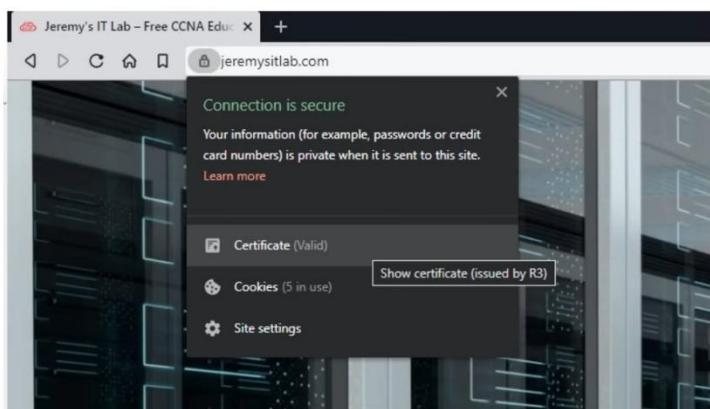
## **Multi-factor Authentication**

- Involves more than just username/password
- Usually involves providing 2 of the following (= 2-factor authentication)
  - Something you know
    - Username/password, PIN, etc
  - Something you have
    - Pressing a notification that appear on your phone, a badge that is scanned, etc

- Something you are
  - Biometrics such as face scan, palm scan, fingerprint scan, etc
- Requiring multiple factors of authentication greatly increases the security
  - Even if attackers learn the target's password, they won't be able to log in to the account

## Digital Certificates

- Another form of authentication used to prove the identity of the holder of the certificate
- Used for websites to verify that the website being accessed is legitimate
- Entities that want a certificate to prove their identity send a CSR (Certificate Signing Request) to a CA (Certificate Authority), which will generate and sign the certificate



## Controlling and Monitoring users with AAA

- AAA (triple-A): Authentication, Authorization, Accounting
- Framework for controlling and monitoring users of a computer system (i.e. a network)
- Authentication
  - Process of verifying a user's identity
  - E.g. Logging in
- Authorization
  - Process of granting the user the appropriate access and permissions
  - E.g. Granting user access to certain files/services, restricting others
- Accounting
  - Process of recording the user's activities on the system
  - E.g. logging when a user makes a change to a file
- Enterprises usually use a AAA server to provide AAA services
  - ISE (Identity Services Engine) is Cisco's AAA server
- AAA servers usually support the following 2 AAA protocols
  - RADIUS

- Open standard protocol
- UDP port 1812 and 1813
- TACACS+
  - Cisco proprietary protocol
  - TCP port 49

## **Security Program Elements**

- User awareness programs
  - Designed to make employees aware of potential security threats and risks
  - E.g. Company send fake phishing emails, those who fall for it will be informed
- User training
  - More formal than user awareness programs
  - E.g. Dedicated training sessions on security policies, how to make strong passwords, etc
- Physical access control
  - Protects equipment and data from potential attackers by only allowing authorized users into protected areas such as network closets or data centre floors
  - Multifactor locks can protect access to restricted areas
  - E.g. Door that requires users to swipe a badge and scan their fingerprint to enter
    - Permissions of the badge can easily be changed

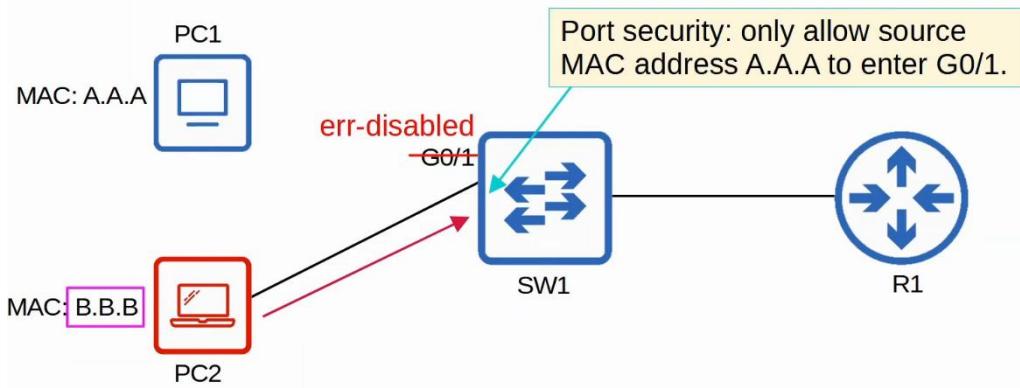
## Port Security

Things covered

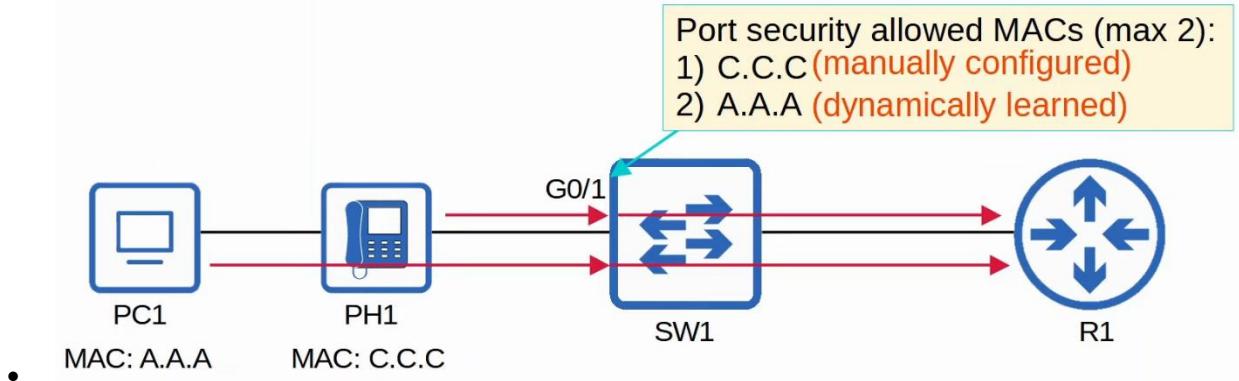
- Intro to port security
- Why use port security
- Config

## **Port security**

- A security feature of Cisco switches
- Allows you to control which source MAC address(es) are allowed to enter the switchport
- If an unauthorized source MAC address enters the port, an action will be taken
  - Default action: place the interface in 'err-disable' state



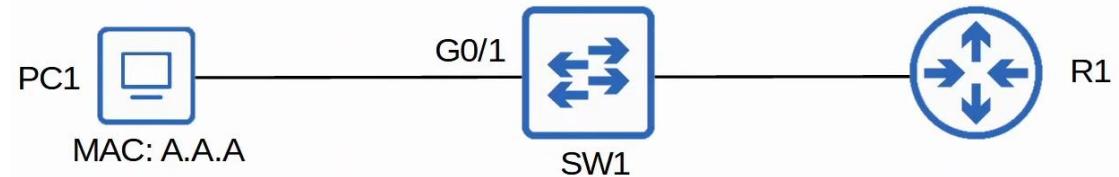
- When you enable port security on an interface with the default settings, 1 MAC address is allowed
  - Can config the allowed MAC address
  - Default: allow the first source MAC address that enters the interface
- Can change the max number of MAC address allowed
  - Can be a combination of manually configured and dynamically learned



## Why port security?

- Allow network admins to control which devices are allowed to access the network
- However, MAC address spoofing is a simple task
  - Easy to configure a device to send frames with a different source MAC address
- Ability to limit the number of MAC address is more useful than specifying MAC addresses allowed on interface
- Think of the DHCP starvation attack
  - Attacker spoofed thousands of fake MAC addresses
  - The DHCP server assigned IP addresses to these fake MAC addresses, exhausting the DHCP pool
  - Switch's MAC address table can also be full in such attacks
- Limiting the number of MAC addresses on an interface can protect against those attacks

## Enabling port security



```

SW1(config)#interface g0/1
SW1(config-if)#switchport port-security
Command rejected: GigabitEthernet0/1 is a dynamic port.

SW1(config-if)#do show int g0/1 switchport
Name: Gi0/1
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: static access
! [output omitted]

SW1(config-if)#switchport mode access

SW1(config-if)#do show int g0/1 switchport
Name: Gi0/1
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access

SW1(config-if)#switchport port-security
SW1(config-if)#
  
```

Port security can be enabled on access ports or trunks ports, but they must be statically configured as access or trunk.  
`switchport mode access = OK`  
`switchport mode trunk = OK`  
`switchport mode dynamic auto`  
`switchport mode dynamic desirable`

The administrative mode is now static access, so the `switchport port-security` command should work.

The command works, so port security is now enabled on G0/1.

```

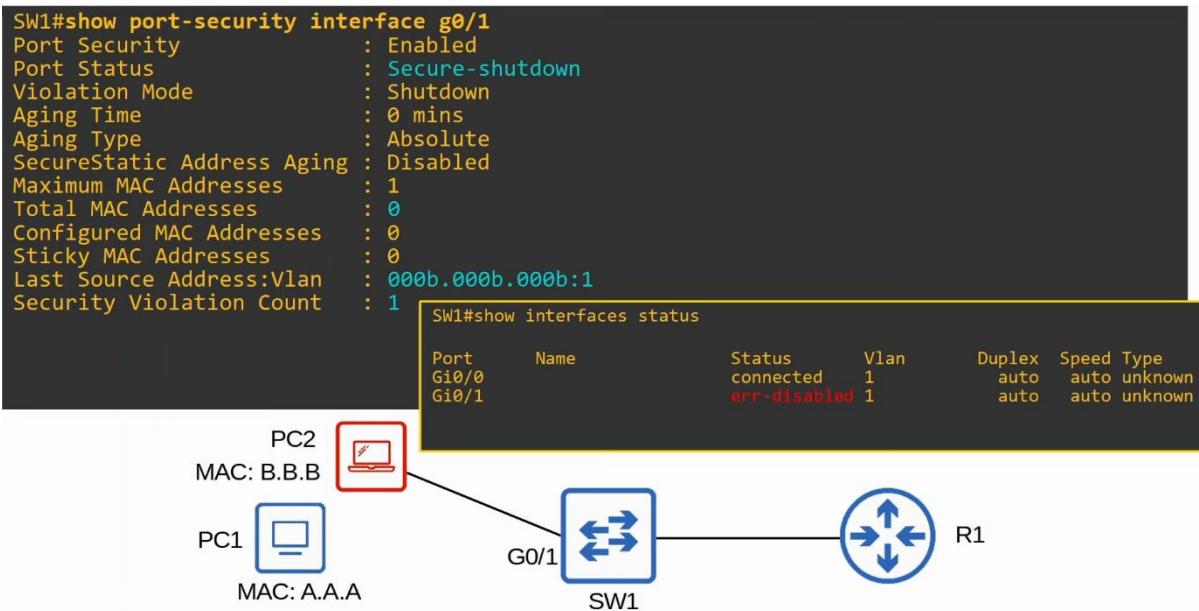
SW1#show port-security interface g0/1
Port Security          : Enabled
Port Status            : Secure-up
Violation Mode        : Shutdown
Aging Time             : 0 mins
Aging Type             : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses : 1
Total MAC Addresses   : 0
Configured MAC Addresses : 0
Sticky MAC Addresses  : 0
Last Source Address:Vlan : 0000.0000.0000:0
Security Violation Count : 0
  
```

- Ping from PC1 to R1

```

SW1#show port-security interface g0/1
Port Security : Enabled
Port Status : Secure-up
Violation Mode : Shutdown
Aging Time : 0 mins
Aging Type : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses : 1
Total MAC Addresses : 1
Configured MAC Addresses : 0
Sticky MAC Addresses : 0
Last Source Address:Vlan : 000a.000a.000a:1
Security Violation Count : 0

```

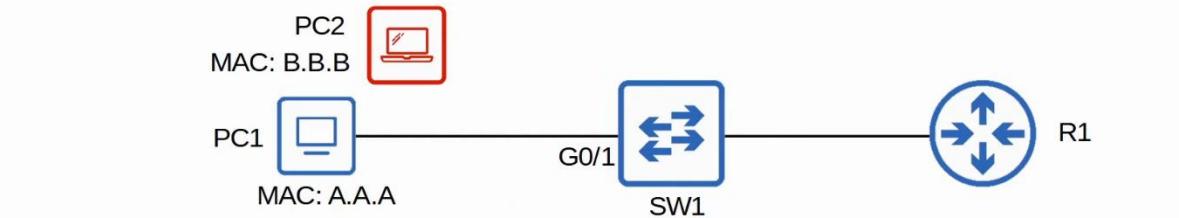


Re-enabling an interface (Manual)

```
SW1(config)#interface g0/1
SW1(config-if)#shutdown
SW1(config-if)#no shutdown
```

- 1) Disconnect the unauthorized device  
2) **shutdown** and then **no shutdown** the interface

```
SW1#show port-security interface g0/1
Port Security           : Enabled
Port Status              : Secure-up
Violation Mode          : Shutdown
Aging Time               : 0 mins
Aging Type               : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses   : 1
Total MAC Addresses      : 0
Configured MAC Addresses: 0
Sticky MAC Addresses     : 0
Last Source Address:Vlan : 0000.0000.0000:0
Security Violation Count: 0
```



### Re-enabling an interface (ErrDisable Recovery)

```
SW1#show errdisable recovery
ErrDisable Reason      Timer Status
-----
arp-inspection         Disabled
bpdu-guard             Disabled
channel-misconfig (STP) Disabled
dhcp-rate-limit        Disabled
dtp-flap               Disabled
! [output omitted due to length]
ps-secure-violation    Disabled
security-violation     Disabled
sfp-config-mismatch    Disabled
storm-control          Disabled
udld                   Disabled
unicast-flood          Disabled
vmps                  Disabled
psp                   Disabled
dual-active-recovery   Disabled
evc-lite input mapping fa Disabled
Recovery command: "clear" Disabled
```

Every 5 minutes (by default), all err-disabled interfaces will be re-enabled **if err-disable recovery has been enabled for the cause of the interface's disablement.**

Timer interval: 300 seconds

Interfaces that will be enabled at the next timeout:

```

SW1(config)#errdisable recovery cause psecure-violation
SW1(config)#errdisable recovery interval 180

SW1#show errdisable recovery
ErrDisable Reason          Timer Status
-----  -----
![output omitted due to length]
psecure-violation      Enabled
![output omitted due to length]

Timer interval: 180 seconds
Interfaces that will be enabled at the next timeout:
Interface      Errdisable reason      Time left(sec)
-----  -----
Gi0/1           psecure-violation      149

```

ErrDisable Recovery is useless if you don't remove the device that caused the interface to enter the err-disabled state!

- Note: If never disconnect the unauthorized device
  - If manual, the interface will just enter err-disabled state again
  - If auto, may select the device as the first device and allow messages from that device, which is bad

## Violation Modes

- There are 3 different violation modes that determine what the switch will do if an unauthorized frame enters an interface configured with port security
  - Shutdown
    - Effectively shuts down the port by placing it in err-disabled state
    - Generates a Syslog and/or SNMP message when the interface is disabled
    - The violation counter is set to 1 when the interface is disabled
  - Restrict
    - The switch discards traffic from unauthorized MAC address
    - Interface NOT disabled
    - Generates a Syslog and/or SNMP message each time an unauthorized MAC is detected
    - Violation counter incremented by 1 for each unauthorized frame
  - Protect
    - Switch discards traffic from unauthorized MAC address
    - Interface NOT disabled
    - Does NOT generate Syslog/SNMP message for unauthorized traffic
    - Does NOT increment counter

Restrict

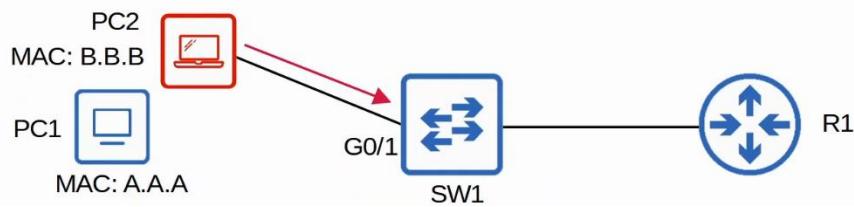
```

SW1(config-if)#switchport port-security
SW1(config-if)#switchport port-security mac-address 000a.000a.000a
SW1(config-if)#switchport port-security violation restrict

*May 23 22:54:09.951: %PORT_SECURITY-2-PSECURE_VIOLATION: Security violation occurred, caused by MAC
address 000b.000b.000b on port GigabitEthernet0/1.

SW1#show port-security interface g0/1
Port Security           : Enabled
Port Status              : Secure-up
Violation Mode          : Restrict
Aging Time               : 0 mins
Aging Type               : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses   : 1
Total MAC Addresses      : 1
Configured MAC Addresses : 1
Sticky MAC Addresses     : 0
Last Source Address:Vlan : 000b.000b.000b:1
Security Violation Count : 12

```



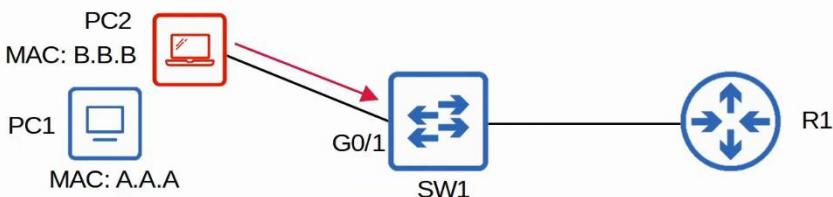
## Protect

```

SW1(config-if)#switchport port-security
SW1(config-if)#switchport port-security mac-address 000a.000a.000a
SW1(config-if)#switchport port-security violation protect

SW1#show port-security interface g0/1
Port Security           : Enabled
Port Status              : Secure-up
Violation Mode          : Protect
Aging Time               : 0 mins
Aging Type               : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses   : 1
Total MAC Addresses      : 1
Configured MAC Addresses : 1
Sticky MAC Addresses     : 0
Last Source Address:Vlan : 000b.000b.000b:1
Security Violation Count : 0

```



## Secure MAC address aging

- Secure MAC address: MAC address learned dynamically / statically configured on port-security enabled port
- By default, secure MAC address will not 'age' out (Aging time : 0 mins)
  - Can be configured with "**switchport port-security aging time minutes**"
- Aging type
  - Absolute (default)
    - After the secure MAC address is learned, the aging timer starts and the MAC address is removed after the timer expires, even if the switch continues receiving frames from the source MAC address
  - Inactivity
    - After the secure MAC address is learned, the aging timer starts but resets every time a frame from that source MAC address is received on the interface
  - Aging type is configured with "**switchport port-security aging type {absolute | inactivity}**"
- Secure Static MAC aging is disabled by default
  - Addresses configured with "**switchport port-security mac-address x.x.x.x**"
  - Can be enabled with "**switchport port-security aging static**"

```
SW1(config-if)#switchport port-security aging time 30
SW1(config-if)#switchport port-security aging type inactivity
SW1(config-if)#switchport port-security aging static

SW1#show port-security interface g0/1
Port Security          : Enabled
Port Status             : Secure-up
Violation Mode          : Shutdown
Aging Time              : 30 mins
Aging Type              : Inactivity
SecureStatic Address Aging : Enabled
Maximum MAC Addresses   : 1
Total MAC Addresses     : 1
Configured MAC Addresses: 1
Sticky MAC Addresses    : 0
Last Source Address:Vlan : 000a.000a.000a:1
Security Violation Count: 0

SW1#show port-security
Secure Port  MaxSecureAddr CurrentAddr  SecurityViolation  Security Action
                  (Count)        (Count)            (Count)
-----
Gi0/1           1             1                 0                Shutdown
-----
Total Addresses in System (excluding one mac per port) : 0
Max Addresses limit in System (excluding one mac per port) : 4096
```

## Sticky Secure MAC addresses

- 'Sticky' secure MAC address learning can be enabled with the following command:
  - SW1(config-if)# **switchport port-security mac-address sticky**
- When enabled, dynamically-learned secure MAC addresses will be added to the running config like this:
  - "**switchport port-security mac-address stick** *mac-address*"
- The sticky secure MAC addresses will never age out
  - You need to save the running-config to the startup-config to make them truly permanent
  - Else, they will not be kept if the switch restarts
- When you issue "**switchport port-security mac-address sticky**" command, all current dynamically learned secure MAC addresses will be converted to sticky secure MAC addresses
- If you issue "**no switchport port-security mac-address sticky**" command, all current sticky secure MAC addresses will be converted to regular dynamically-learned secure MAC addresses

```
SW1(config-if)#switchport port-security
SW1(config-if)#switchport port-security mac-address sticky
SW1(config-if)#do show running-config interface g0/1
!
interface GigabitEthernet0/1
  switchport mode access
  switchport port-security mac-address sticky
  switchport port-security mac-address sticky 000a.000a.000a
  switchport port-security
    negotiation auto
```

## MAC Address Table

- Secure MAC addresses will be added to the MAC address table like any other MAC address
  - Sticky and Static secure MAC addresses will have a type of STATIC
  - Dynamically-learned secure MAC addresses will have a type of DYNAMIC
  - Can view all secure MAC addresses with "**show mac address-table secure**"

```

SW1#show mac address-table secure
      Mac Address Table
-----
Vlan      Mac Address          Type      Ports
----      -----
  1        000a.000a.000a    STATIC    Gi0/1
Total Mac Addresses for this criterion: 1

```

- The type is STATIC even though it is learned dynamically

## Summary

```

SW1# show mac address-table secure
SW1# show port-security
SW1# show port-security interface interface
SW1# show errdisable recovery
SW1(config)# errdisable recovery cause psecure-violation
SW1(config)# errdisable recovery interval seconds
SW1(config-if)# switchport port-security
SW1(config-if)# switchport port-security mac-address mac-address
SW1(config-if)# switchport port-security mac-address sticky
SW1(config-if)# switchport port-security violation {shutdown | restrict | protect}
SW1(config-if)# switchport port-security aging time minutes
SW1(config-if)# switchport port-security aging type {absolute | inactivity}
SW1(config-if)# switchport port-security aging static

```

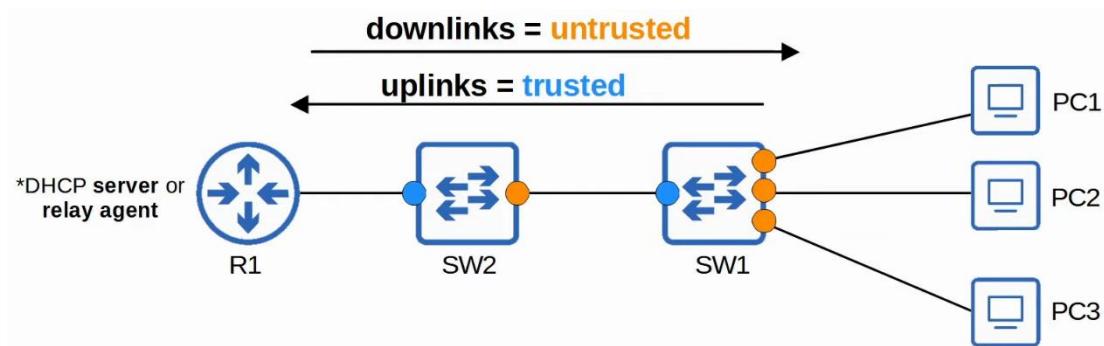
## DHCP Snooping

### Things covered

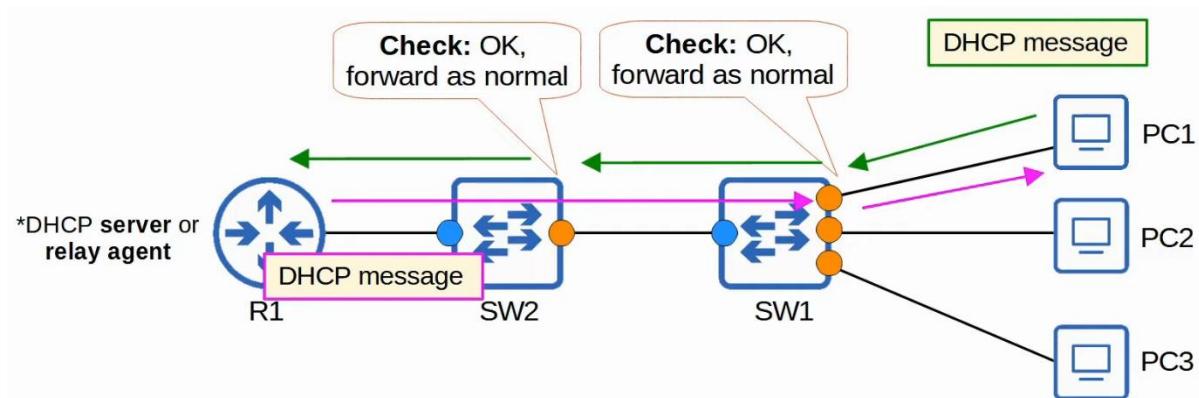
- What is DHCP snooping
- How does it work
- What attacks does it prevent
- Config

## DHCP Snooping

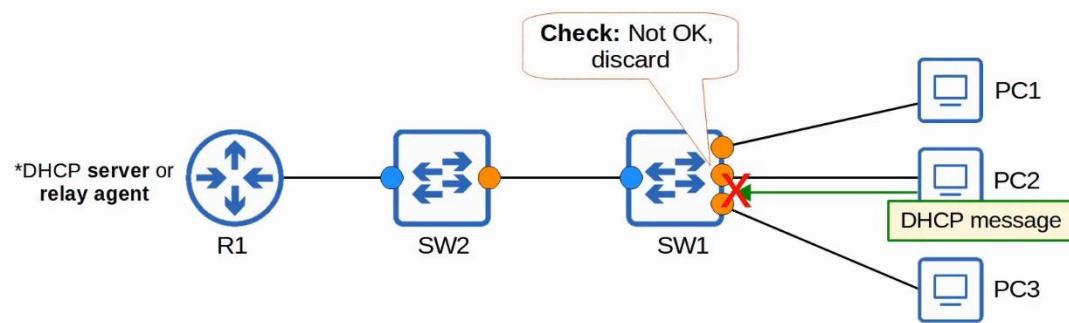
- A security feature of switches that is used to filter DHCP messages received on untrusted ports
- DHCP snooping only filters DHCP messages
  - Non-DHCP are not affected
- All ports are untrusted by default
  - Usually, uplink ports are configured as trusted ports, and downlink ports remain untrusted



- Can trust that R1 won't do anything bad, but cannot trust PCs won't do anything bad



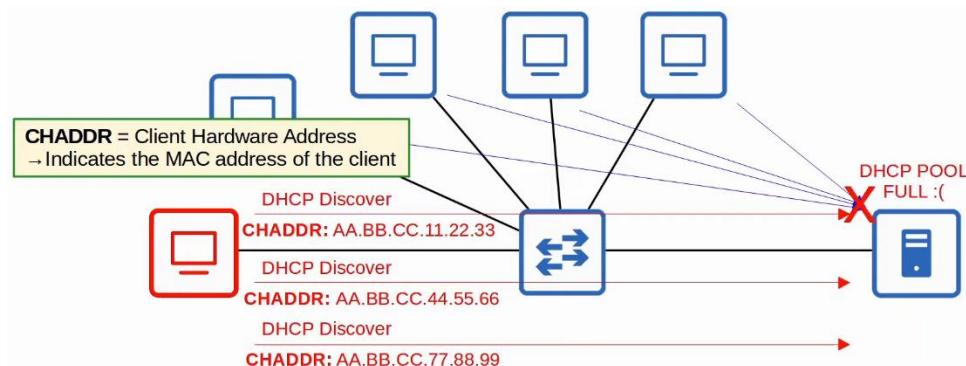
- DHCP messages are checked when arrive on untrusted ports
- Not checked if arrive on trusted ports



### What it protects

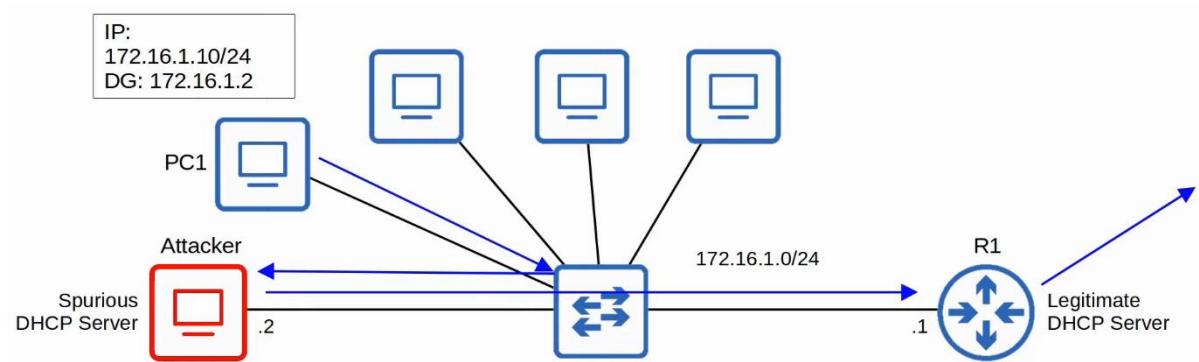
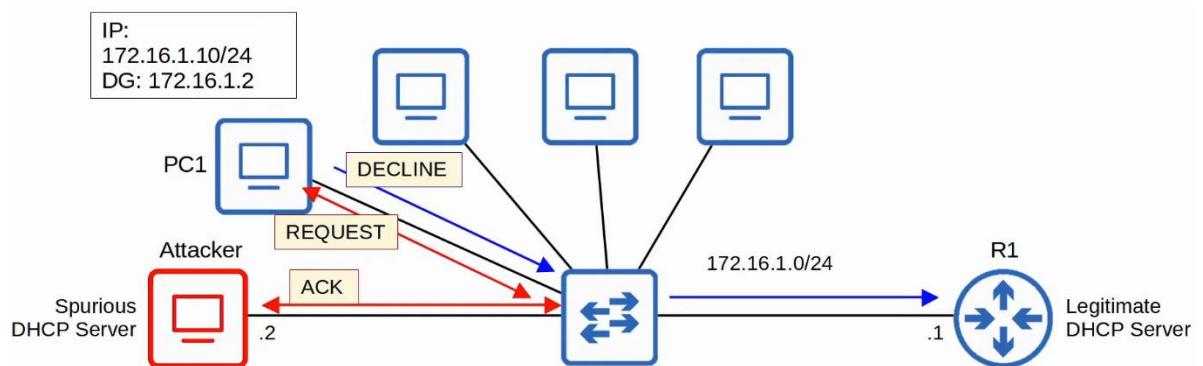
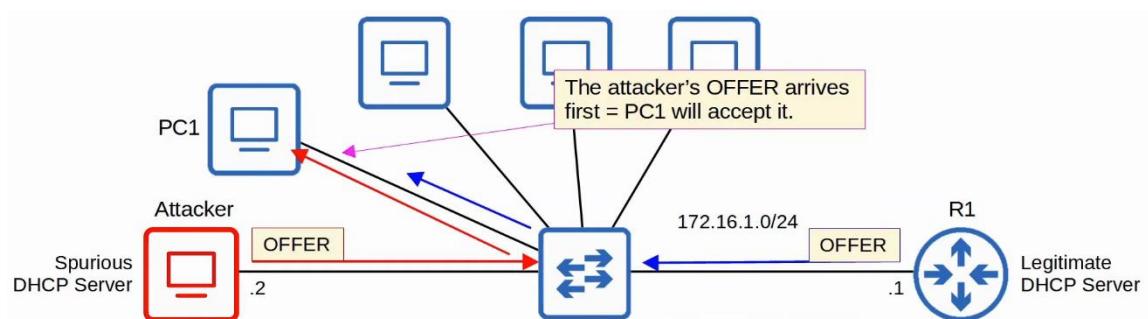
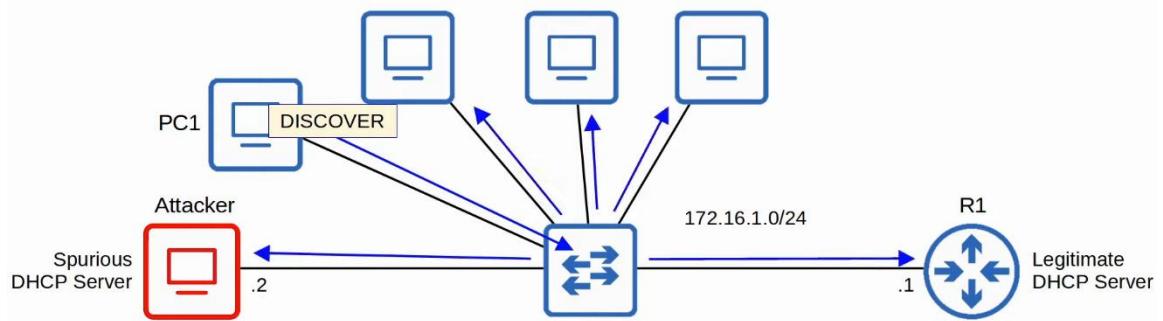
#### DHCP Starvation

- An example of DHCP-based attacks is DHCP starvation attack
- An attacker uses spoofed MAC addresses to flood DHCP Discover messages
- The target server's DHCP pool becomes full, resulting in a DoS to other devices



#### DHCP Poisoning (Man-in-the-middle)

- Similar to ARP poisoning, DHCP poisoning can also be used to perform Man-in-the-middle attack
- A spurious DHCP server replies to clients' DHCP Discover messages and assigns them IP addresses, but makes the clients use the spurious server's IP as the default gateway
  - \*Clients usually accept the first Offer message they receive
  - If router is only a DHCP relay, then more likely PC will use the attacker's Offer message
- This will cause the client to send traffic to the attacker instead of the legitimate default gateway
- The attacker can examine/modify the traffic before forwarding it to the actual default gateway



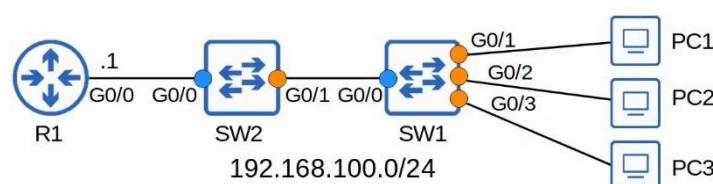
## DHCP Messages

- When DHCP Snooping filters messages, it differentiates btw DHCP Server and DHCP Client messages
- Messages sent by DHCP Servers
  - OFFER
  - ACK
  - NAK (Opposite of ACK, used to decline a client's REQUEST)
- Messages sent by DHCP Clients
  - DISCOVER
  - REQUEST
  - RELEASE (used to tell the server that the client no longer need its IP address)
  - DECLINE (used to decline the IP address offered by a DHCP server)

## DHCP Snooping Operations

- If a DHCP message received on trusted ports, forward it as normal w/o inspection
- IF a DHCP message received on untrusted ports, inspect it and act as follows
  - If a DHCP Server message, discard it
  - If a DHCP Client message, perform the following checks
    - DISCOVER/REQUEST messages
      - Check if the frame's source MAC address and DHCP message's CHADDR fields match
      - Match = forward, mismatch = drop
    - RELEASE/DECLINE messages
      - Check if the packet's source IP address and the receiving interface match the entry in the DHCP Snooping Binding Table
      - Match = forward, mismatch = discard
- When a client successfully leases an IP address from a server, create a new entry in the DHCP Snooping Binding Table

## Config



```
SW2(config)#ip dhcp snooping
SW2(config)#ip dhcp snooping vlan 1
SW2(config)#no ip dhcp snooping information option-
SW2(config)#interface g0/0
SW2(config-if)#ip dhcp snooping trust
```

```

SW1(config)#ip dhcp snooping
SW1(config)#ip dhcp snooping vlan 1
SW1(config)#no ip dhcp snooping information option
SW1(config)#interface g0/0
SW1(config-if)#ip dhcp snooping trust

```

RELEASE/DECLINE messages will be checked to make sure their IP address/interface ID match the entry in the DHCP snooping table.

MacAddress	IpAddress	Lease(sec)	Type	VLAN	Interface
0C:29:2F:18:79:00	192.168.100.10	86294	dhcp-snooping	1	GigabitEthernet0/3
0C:29:2F:90:91:00	192.168.100.11	86302	dhcp-snooping	1	GigabitEthernet0/1
0C:29:2F:67:E9:00	192.168.100.12	86314	dhcp-snooping	1	GigabitEthernet0/2
Total number of bindings: 3					

## DHCP Snooping Rate-Limiting

- DHCP Snooping can limit the rate at which DHCP messages are allowed to enter an interface
- If the rate of DHCP messages crosses the configured limit, the interface is err-disabled
- Like with Port Security, the interface can be manually re-enabled, or automatically re-enabled with errdisable recovery
- Rate-limiting can be very useful to protect against DHCP exhaustion attacks

```

SW1(config)#interface range g0/1 - 3
SW1(config-if-range)#ip dhcp snooping limit rate 1

*Jun  5 13:15:14.180: %DHCP_SNOOPING-4-DHCP_SNOOPING_ERRDISABLE_WARNING: DHCP Snooping received 1 DHCP packets on
interface Gi0/1
*Jun  5 13:15:14.181: %DHCP_SNOOPING-4-DHCP_SNOOPING_RATE_LIMIT_EXCEEDED: The interface Gi0/1 is receiving more
than the threshold set
*Jun  5 13:15:14.182: %PM-4-ERR_DISABLE: dhcp-rate-limit error detected on Gi0/1, putting Gi0/1 in err-disable
state
*Jun  5 13:15:15.185: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to down
*Jun  5 13:15:16.190: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to down

```

```

SW1(config)#errdisable recovery cause dhcp-rate-limit

SW1#show errdisable recovery
ErrDisable Reason          Timer Status
-----                    -----
arp-inspection             Disabled
bpduguard                  Disabled
channel-misconfig (STP)    Disabled
dhcp-rate-limit            Enabled
dtp-flap                   Disabled
gbic-invalid               Disabled
inline-power                Disabled
![output omitted due to length]

Timer interval: 300 seconds

Interfaces that will be enabled at the next timeout:

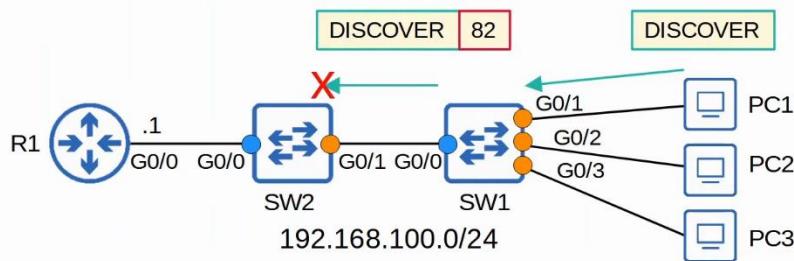
Interface      Errdisable reason      Time left(sec)
-----          -----                -----
Gi0/1          dhcp-rate-limit        293

```

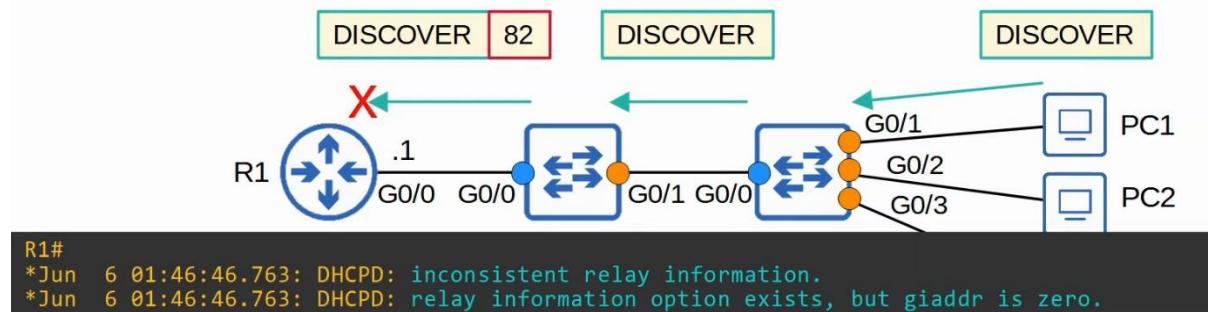
## DHCP Option 82 (Information Option)

- Option 82, also known as the "DHCP relay agent information option" is one of the many DHCP options
- It provides additional information about which DHCP relay agent received the client's message, on which interface, in which VLAN, etc
- DHCP relay agents can add Option 82 to messages they forward to remote DHCP servers
- With DHCP Snooping enabled, by default, Cisco switches will add Option 82 to DHCP messages they receive from clients, even if the switch is not acting as a DHCP relay agent
- By default, Cisco switches will drop DHCP messages with Option 82 that are received on untrusted ports

```
SW2#
*Jun 6 01:36:15.298: %DHCP_SNOOPING-5-DHCP_SNOOPING_NONZERO_GIADDR: DHCP_SNOOPING drop message with non-zero giaddr or option82 value on untrusted port, message type: DHCPDISCOVER, MAC sa: 0c29.2f67.e900
```



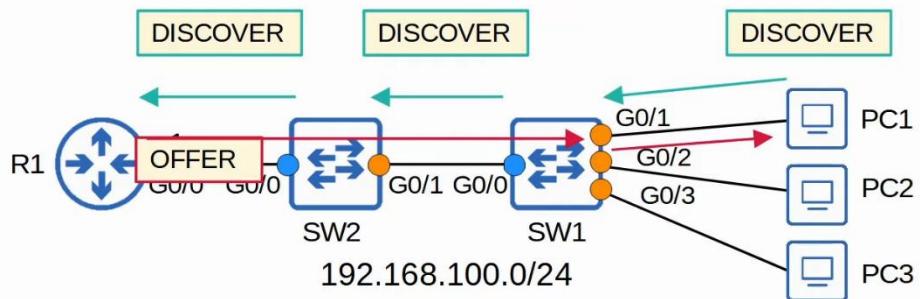
```
SW1(config)#no ip dhcp snooping information option
```



- "inconsistent relay information": received Option 82 from SW2 even though not a DHCP relay agent

```
SW1(config)#no ip dhcp snooping information option
```

```
SW2(config)#no ip dhcp snooping information option
```



## Summary

```
SW1(config)# ip dhcp snooping
SW1(config)# ip dhcp snooping vlan vlan-number
SW1(config)# errdisable recovery cause dhcp-rate-limit
SW1(config)# no ip dhcp snooping information option
SW1(config-if)# ip dhcp snooping trust
SW1(config-if)# ip dhcp snooping limit rate packets-per-second
SW1# show ip dhcp snooping binding
```

## Dynamic ARP Inspection

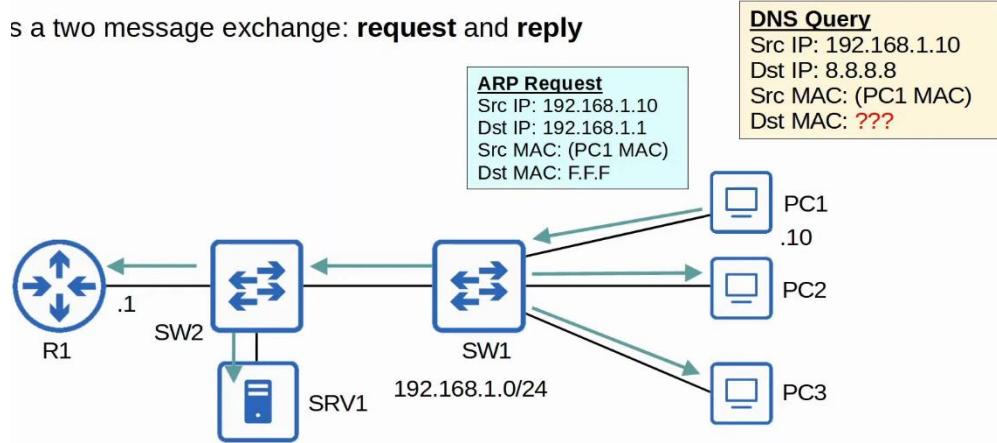
### Things covered

- What is Dynamic ARP Inspection
- How does it work
- What attack does it prevent
- Config

## ARP Review

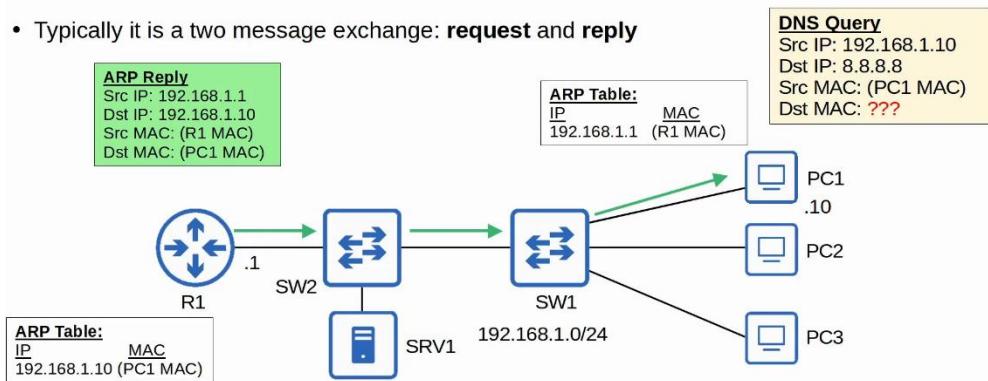
- ARP is used to learn the MAC address of another device with a known IP address
- E.g. a PC will use ARP to learn the MAC address of its default gateway to communicate with external networks
- Typically, it is a 2 message exchange: request, reply

is a two message exchange: **request** and **reply**



```
> Frame 99: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
✓ Ethernet II, Src: 0c:29:2f:90:91:00 (0c:29:2f:90:91:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  > Source: 0c:29:2f:90:91:00 (0c:29:2f:90:91:00)
    Type: ARP (0x0806)
    Padding: 0000000000000000000000000000000000000000000000000000000000000000
  ✓ Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: 0c:29:2f:90:91:00 (0c:29:2f:90:91:00)
    Sender IP address: 192.168.1.10
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.1.1
```

- Typically it is a two message exchange: **request** and **reply**

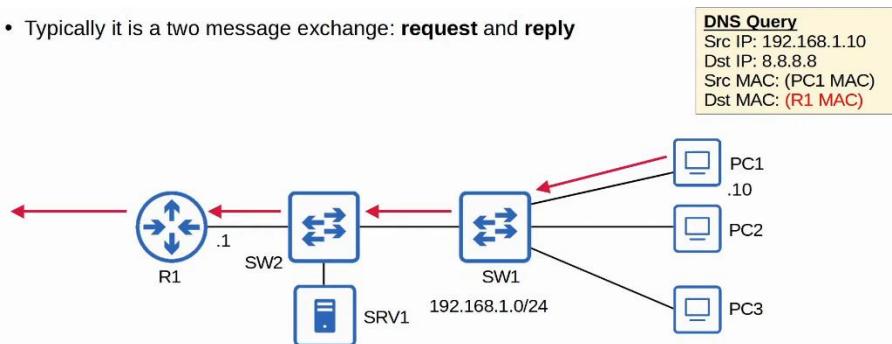


```

> Frame 224: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
▼ Ethernet II, Src: 0c:29:2f:43:b5:00 (0c:29:2f:43:b5:00), Dst: 0c:29:2f:90:91:00 (0c:29:2f:90:91:00)
  > Destination: 0c:29:2f:90:91:00 (0c:29:2f:90:91:00)
  > Source: 0c:29:2f:43:b5:00 (0c:29:2f:43:b5:00)
  Type: ARP (0x0806)
  Padding: 0000000000000000000000000000000000000000000000000000000000000000
▼ Address Resolution Protocol (reply)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: reply (2)
  Sender MAC address: 0c:29:2f:43:b5:00 (0c:29:2f:43:b5:00)
  Sender IP address: 192.168.1.1
  Target MAC address: 0c:29:2f:90:91:00 (0c:29:2f:90:91:00)
  Target IP address: 192.168.1.10

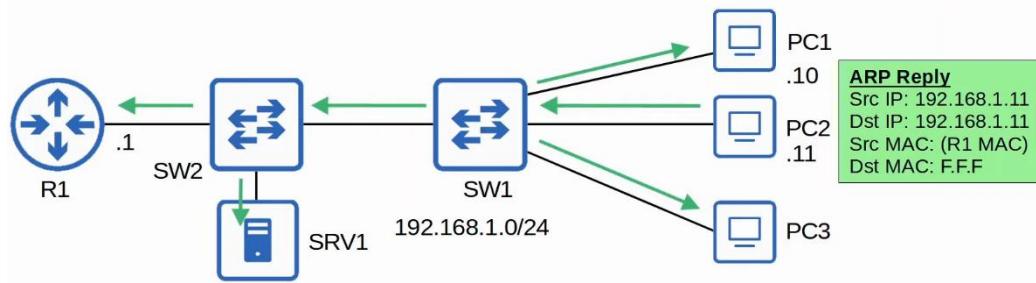
```

- Typically it is a two message exchange: **request** and **reply**



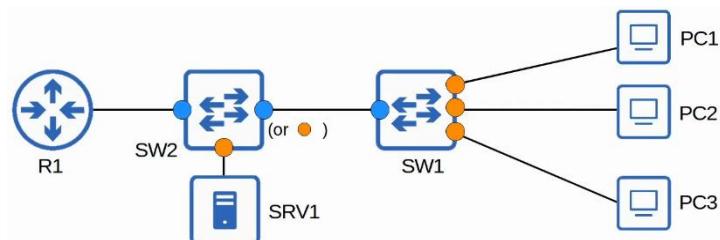
## Gratuitous ARP

- It is an ARP reply that is sent w/o receiving an ARP Request
- It is sent to the broadcast MAC address
- It allows other devices to learn the MAC address of the sending device w/o having to send ARP requests
- Some devices automatically send GARP messages when an interface is enabled, IP address is changed, MAC address is changed, etc

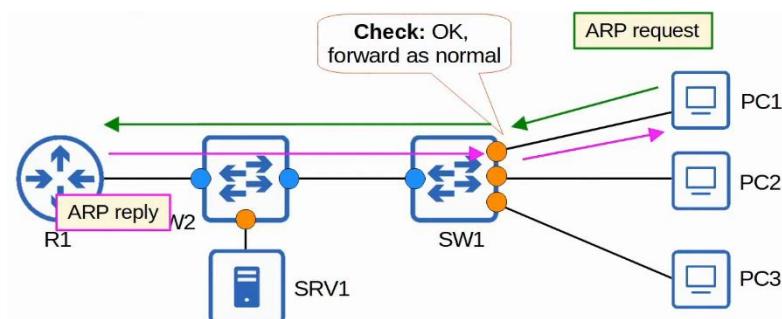


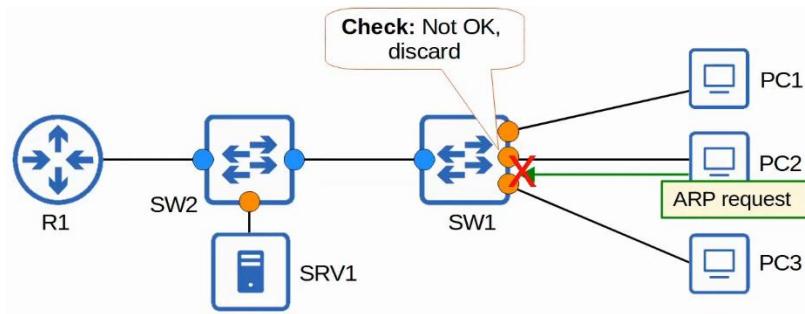
### Dynamic ARP Inspection

- DAI is a security feature of switches that is used to filter ARP messages received on untrusted ports
- DAI only filters ARP messages
  - Non-ARP messages are not affected
- All ports are untrusted by default
  - Typically, all ports connected to other network devices (switches, routers) should be configured as trusted
  - Interfaces connected to end hosts should remain untrusted



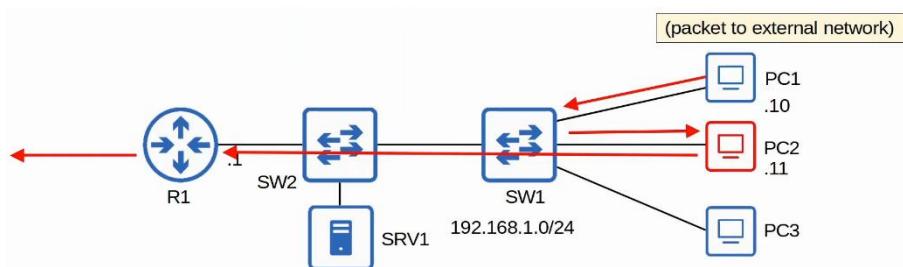
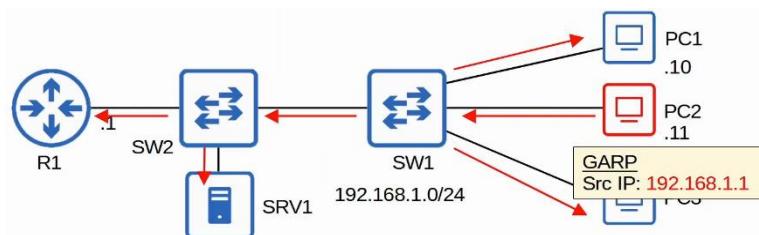
- SW2 interface facing downlink can also be untrusted
  - But Cisco documentation say any ports connected to network devices should be trusted





### ARP Poisoning (Man-in-the-middle)

- Similar to DHCP poisoning, ARP poisoning involves an attacker manipulating targets' ARP tables so traffic is sent to the attacker
- To do this, the attacker can send gratuitous ARP messages using another device's IP address
- Other devices in the network will receive the GARP and update their ARP table, causing them to send traffic to the attacker instead of the legitimate destination



### DAI Operations

- DAI inspects the sender MAC and sender IP fields of ARP messages received on untrusted ports and checks that there is a matching entry in the DHCP snooping binding table
  - If there is a matching entry, the ARP message is forwarded normally

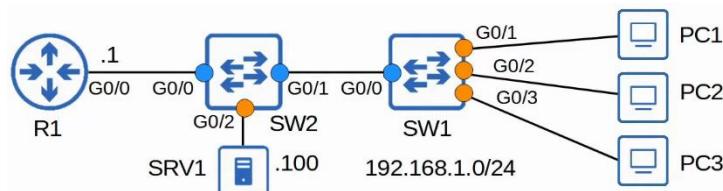
```

SW1#show ip dhcp snooping binding
MacAddress          IPAddress      Lease(sec) Type        VLAN   Interface
-----              -----          -----       -----      -----   -----
0C:29:2F:18:79:00  192.168.100.10  86294     dhcp-snooping 1    GigabitEthernet0/3
0C:29:2F:90:91:00  192.168.100.11  86302     dhcp-snooping 1    GigabitEthernet0/1
0C:29:2F:E9:00:00  192.168.100.12  86314     dhcp-snooping 1    GigabitEthernet0/2
Total number of bindings: 3

```

- DAI does not inspect messages received on trusted ports
  - They are forwarded as normal
- ARP ACLs can be manually configured to map IP addresses/MAC addresses for DAI to check
  - Useful for hosts that don't use DHCP
- DAI can be configured to perform more in depth check
- DAI also supports rate-limiting to prevent attackers from overwhelming the switch with ARP messages
  - DHCP snooping and DAI both require work from the switch's CPU
  - Even if the attacker's messages are blocked, they can overload the CPU with ARP messages

## Config



```

SW2(config)#ip arp inspection vlan 1
SW2(config)#interface range g0/0 - 1
SW2(config-if-range)#ip arp inspection trust

```

```

SW1(config)#ip arp inspection vlan 1
SW1(config)#interface g0/0
SW1(config-if)#ip arp inspection trust

```

- Don't need to configure globally like in DHCP snooping, just straight away on VLAN

"show ip arp inspection interfaces"

Interface	Trust State	Rate (pps)	Burst Interval
Gi0/0	Trusted	None	N/A
Gi0/1	Untrusted	15	1
Gi0/2	Untrusted	15	1
Gi0/3	Untrusted	15	1
Gi1/0	Untrusted	15	1
Gi1/1	Untrusted	15	1
Gi1/2	Untrusted	15	1
Gi1/3	Untrusted	15	1
Gi2/0	Untrusted	15	1
Gi2/1	Untrusted	15	1
Gi2/2	Untrusted	15	1
Gi2/3	Untrusted	15	1
Gi3/0	Untrusted	15	1
Gi3/1	Untrusted	15	1
Gi3/2	Untrusted	15	1
Gi3/3	Untrusted	15	1

## DAI rate-limiting

```
SW1(config)#interface range g0/1 - 2
SW1(config-if-range)#ip arp inspection limit rate 25 burst interval 2
SW1(config-if-range)#interface range g0/3
SW1(config-if)#ip arp inspection limit rate 10
SW1(config-if)#do show ip arp inspection interfaces
```

Interface	Trust State	Rate (pps)	Burst Interval
Gi0/0	Trusted	None	N/A
Gi0/1	Untrusted	25	2
Gi0/2	Untrusted	25	2
Gi0/3	Untrusted	10	1

The burst interval is optional. If you don't specify it, the default is 1 second.

If ARP messages are received faster than the specified rate, the interface will be err-disabled. It can be re-enabled in two ways:  
 1: shutdown and no shutdown  
 2: errdisable recovery cause arp-inspection

## DAI Optional Checks

```
SW1(config)#ip arp inspection validate ?
  dst-mac  Validate destination MAC address
  ip      Validate IP addresses
  src-mac Validate source MAC address
```

- These checks are done in addition to the standard DAI check (sender MAC/IP)
  - If configured, an ARP message must pass all of the checks to be considered valid
- "dst-mac"
  - Check if destination MAC address in Ethernet header and ARP body are the same
- "ip"
  - Check for invalid and unexpected IP address
    - E.g. 0.0.0.0, 255.255.255.255, all IP multicast address
  - Check the sender IP in all ARP request and response
  - Check the target IP only in ARP response
- "src-mac"
  - Check if source MAC address in Ethernet header and ARP body are the same

```

> Frame 224: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
  ▾ Ethernet II, Src: 0c:29:2f:43:b5:00 (0c:29:2f:43:b5:00), Dst: 0c:29:2f:90:91:00 (0c:29:2f:90:91:00)
    ▾ Destination: 0c:29:2f:90:91:00 (0c:29:2f:90:91:00)
    ▾ Source: 0c:29:2f:43:b5:00 (0c:29:2f:43:b5:00)
      Type: ARP (0x0806)
      Padding: 0000000000000000000000000000000000000000000000000000000000000000
  ▾ Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: 0c:29:2f:43:b5:00 (0c:29:2f:43:b5:00)
    Sender IP address: 192.168.1.1
    Target MAC address: 0c:29:2f:90:91:00 (0c:29:2f:90:91:00)
    Target IP address: 192.168.1.10

```

```

SW1(config)#ip arp inspection validate dst-mac
SW1(config)#ip arp inspection validate ip
SW1(config)#ip arp inspection validate src-mac

SW1(config)#do show running-config | include validate
ip arp inspection validate src-mac

SW1(config)#ip arp inspection validate ip src-mac dst-mac

SW1(config)#do show running-config | include validate
ip arp inspection validate src-mac dst-mac ip

```

- Must enter all the validation checks in a single command
  - Can specify 1, 2 or 3
  - The order does not matter

## ARP ACLs

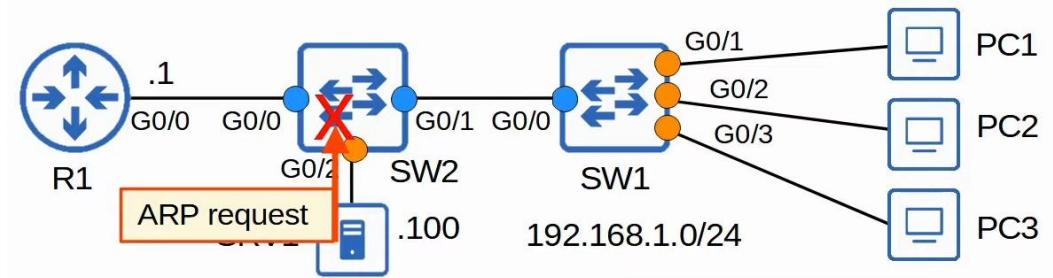
```

SW2#show ip dhcp snooping binding
MacAddress          IPAddress        Lease(sec)  Type           VLAN   Interface
-----  -----
0C:29:2F:18:79:00  192.168.1.12    79226      dhcp-snooping  1      GigabitEthernet0/1
0C:29:2F:90:91:00  192.168.1.10    79188      dhcp-snooping  1      GigabitEthernet0/1
0C:29:2F:67:E9:00  192.168.1.11    79210      dhcp-snooping  1      GigabitEthernet0/1
Total number of bindings: 3

!SRV1 has a static IP address of 192.168.1.100, so it does not have an entry in SW2's DHCP
!snooping binding table.

*Jun 19 05:56:15.538: %SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on Gi0/2, vlan 1.
([0c29.2f1e.7700/192.168.1.100/0000.0000.0000/192.168.1.1/05:56:14 UTC Sat Jun 19 2021])

```



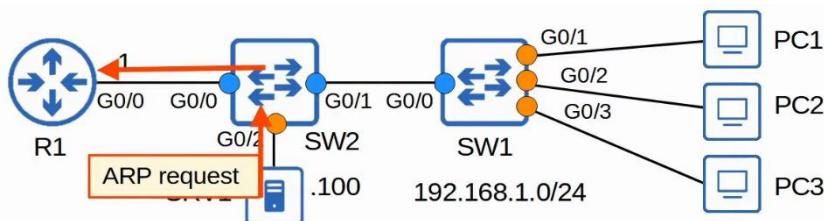
- SRV1 not in SW2 DHCP snooping binding table

```

SW2(config)#arp access-list ARP-ACL-1
SW2(config-arp-nacl)#permit ip host 192.168.1.100 mac host 0c29.2f1e.7700

SW2(config)#ip arp inspection filter ARP-ACL-1 vlan 1

```



"show ip arp inspection"

```

SW2#show ip arp inspection
Source Mac Validation : Enabled
Destination Mac Validation : Enabled
IP Address Validation : Enabled

Vlan Configuration Operation ACL Match Static ACL
--- -----
 1 Enabled Active ARP-ACL-1 No

Vlan ACL Logging DHCP Logging Probe Logging
--- -----
 1 Deny Deny Off

Vlan Forwarded Dropped DHCP Drops ACL Drops
--- -----
 1 56 4 4 0

Vlan DHCP Permits ACL Permits Probe Permits Source MAC Failures
--- -----
 1 0 1 0 0

Vlan Dest MAC Failures IP Validation Failures Invalid Protocol Data
--- -----
 1 0 0 0 0

```

- If static ACL is set to yes, the implicit deny at the end of the ARP ACL will take effect.
- This will cause all ARP messages not permitted by the ARP ACL to be denied.
- In effect, this means that only the ARP ACL will be checked, the DHCP snooping table will not be checked.

## Summary

```
SW1(config)# ip arp inspection vlan vlan-number
SW1(config)# errdisable recovery cause arp-inspection
SW1(config)# ip arp inspection validate (src-mac | dst-mac | ip)
SW1(config-if)# ip arp inspection trust
SW1(config-if)# ip arp inspection limit rate packets [burst interval seconds]

SW1(config)# arp access-list name
SW1(config-arp-nacl)# permit ip host ip-address mac host mac-address
SW1(config)# ip arp inspection filter arp-acl-name vlan vlan-number

SW1# show ip arp inspection
SW1# show ip arp inspection interfaces
```

# Network Architectures

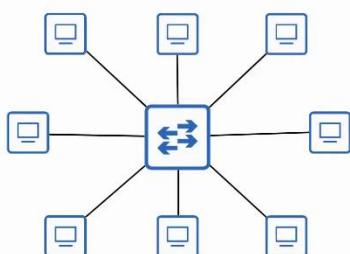
## LAN Architectures

### Things covered

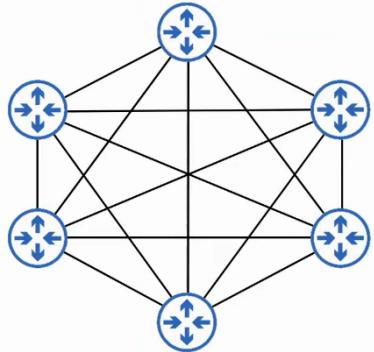
- 2-tier and 3-tier LAN Architecture
- Spine-leaf architecture
- SOHO (Small Office / Home office)

### Common Terminologies

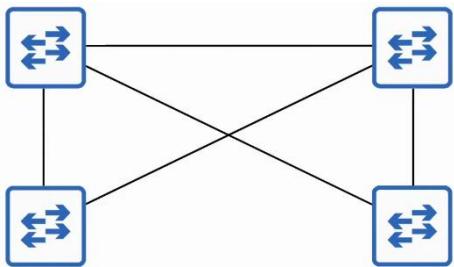
- Star
  - When several devices all connect to 1 central device



- Full Mesh
  - When each device is connected to all other device



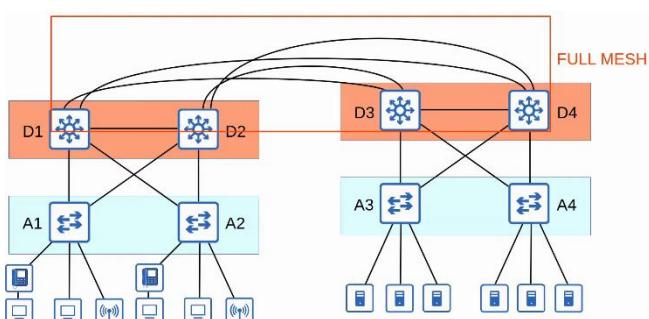
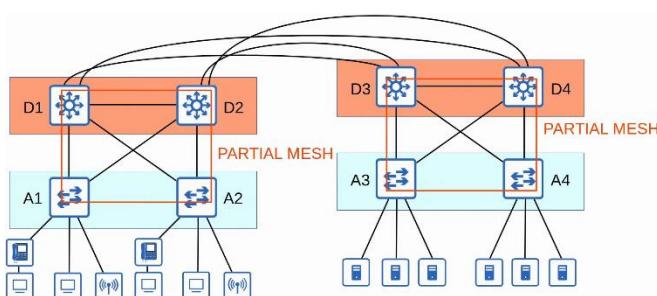
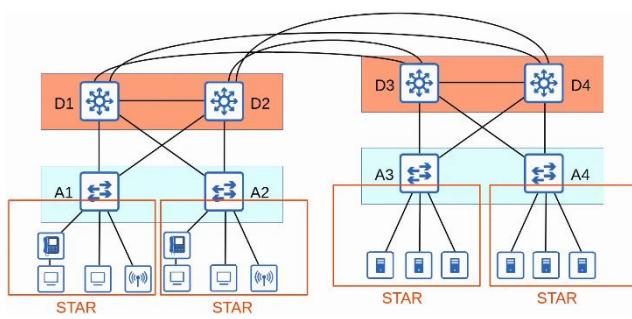
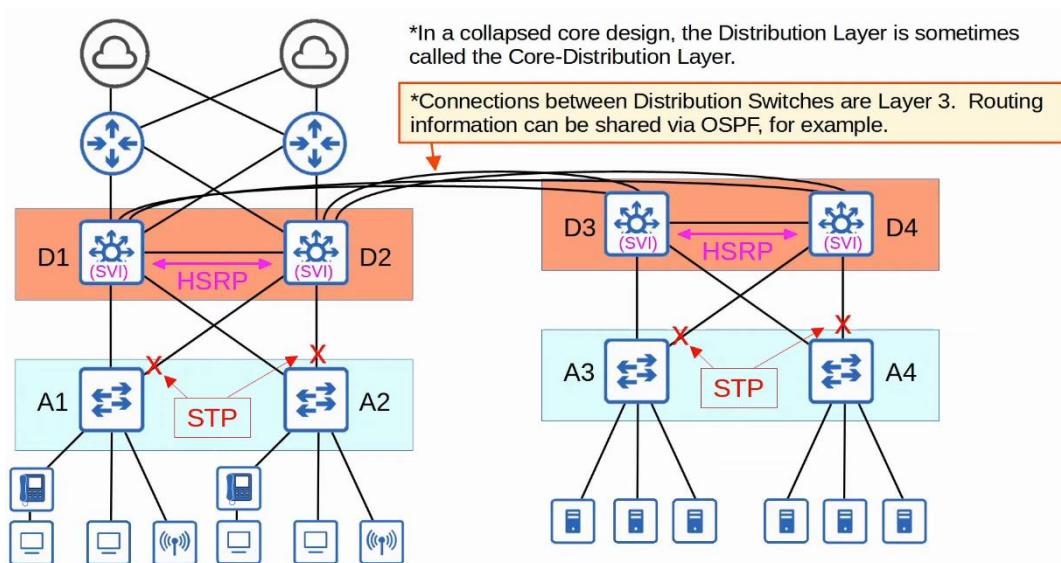
- Partial Mesh
  - When only some devices are connected to all other devices



## 2-Tier Campus LAN Design

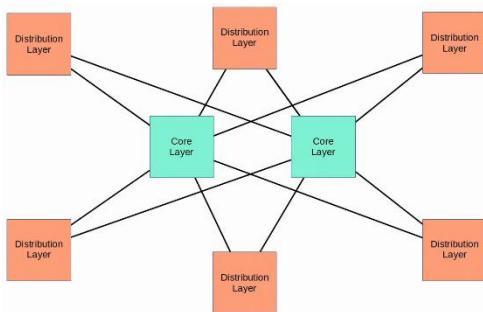
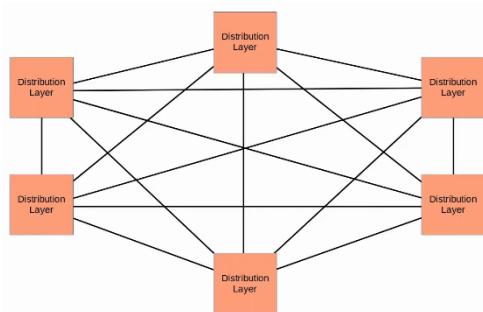
- The 2-tier LAN design consists of 2 hierarchical layers
  - Access Layer
  - Distribution Layer
- Also called a "Collapsed Core" design because it omits a layer that is found in the 3-tier design, the Core Layer
- **Access Layer**
  - The layer that end hosts connects to (PCs, printers, cameras, etc)
  - Typically Access Layer switches have lots of ports for end hosts to connect to
  - QoS marking is typically done here (good to mark frames ASAP)
  - Security services like port security, DAI, etc are typically performed here
  - Switchports might be PoE-enabled for wireless APs, IP Phones, etc
- **Distribution Layer (Aggregation Layer)**
  - Aggregates connections from the Access Layer Switches
  - Typically is the border btw Layer 2 and Layer 3
    - They run both services, e.g. OSPF etc
  - Connects to services such as Internet, WAN, etc (\*for 2 tier design)

## Example



## Problem

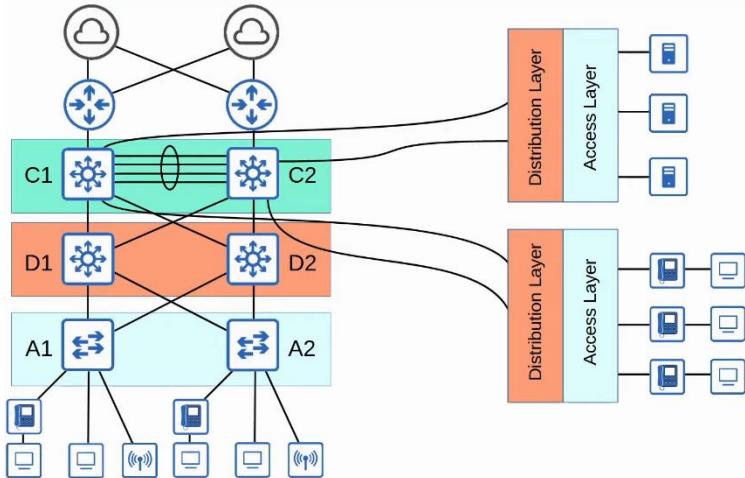
- In large LAN networks with many Distribution Layer switches (e.g. in separate buildings), the number of connections required btw Distribution Layer switches grows rapidly
- To help scale large LAN networks, a Core Layer can be added
  - Cisco recommends adding a Core Layer if there are more than 3 Distribution Layers in a single location



## 3-Tier Campus LAN Design

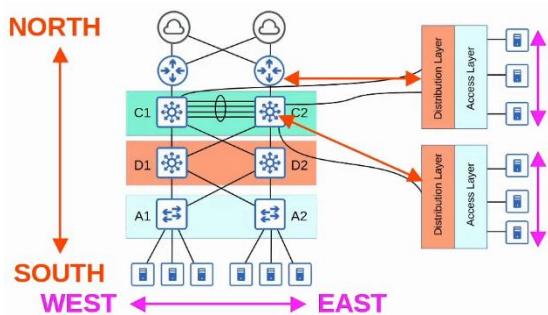
- Consist of 3 hierarchical layers
  - Access Layer
  - Distribution Layer
  - Core Layer
- **Core Layer**
  - Connects Distribution Layers together in large LAN networks
  - The focus is speed ('fast transport') - only care about forwarding packets from 1 Distribution Layer switch to another
  - CPU-intensive operations such as security, QoS marking/classification, etc should be avoided at this Layer
  - Connections are all Layer 3, no spanning-tree

- Should maintain connectivity throughout the LAN even if devices fail



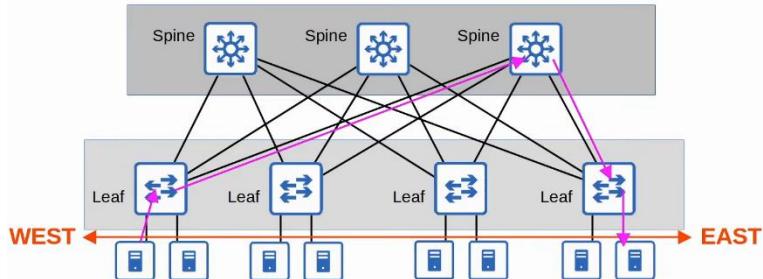
### Spine-Leaf Architecture

- Data centres are dedicated spaces/buildings used to store computer systems such as servers and network devices
- Traditional data centre design uses a 3-tier architecture (Access-Distribution-Core)
- This worked well when most traffic in the data centre was North-South



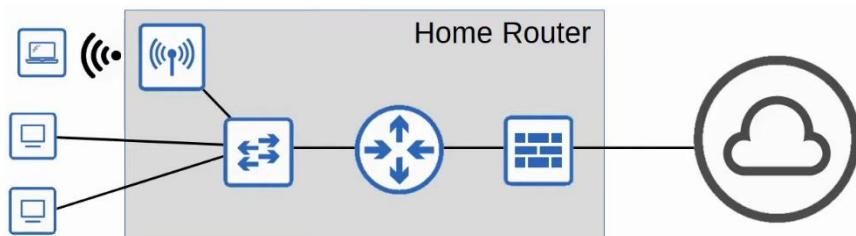
- With the precedence of virtual servers, applications are often deployed in a distributed manner (across multiple physical servers), which increases the amount of East-West traffic in the data centre
- The traditional 3-tier architecture led to bottlenecks in bandwidth as well as variability in the server-to-server latency depending on the path the traffic takes
- To solve this, Spine-Leaf architecture (Clos architecture) has become prominent in data centres
- There are some rules about Spine-Leaf architecture
  - Every leaf switch is connected to every spine switch
  - Every spine switch is connected to every Leaf switch
  - Leaf switches do not connect to other Leaf switches
  - Spine switches do not connect to other Spine switches

- End hosts (servers, etc) only connect to Leaf switches
- The path taken is randomly chosen to balance the traffic load among Spine switches
- Each server is separated by the same number of 'hops' (except those connected by the same Leaf switch), providing consistent latency for East-West traffic
- Easy to scale



## SOHO Networks

- Small Office / Home Office (SOHO) refers to the office of a small company, or a small home office with few devices
  - Does not have to be an actual home 'office', if your home has a network connected to the Internet it is considered a SOHO network
- SOHO networks don't have complex needs, so all networking functions are typically provided by a single device, often called a 'home router' or 'wireless router'
- This 1 device can serve as a
  - Router
  - Switch
  - Firewall
  - Wireless Access Point
  - Modem



## WAN Architectures

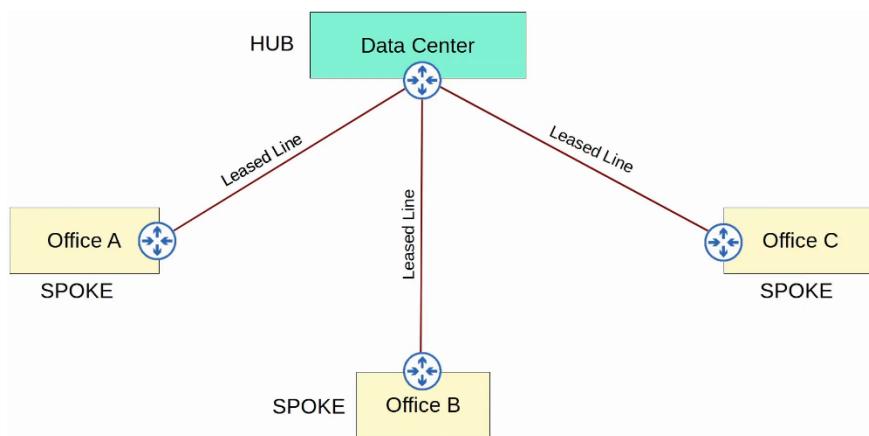
Things covered

- Intro to WANs
- Leased Lines
- MPLS VPNs
- Internet connectivity
- Internet VPNs

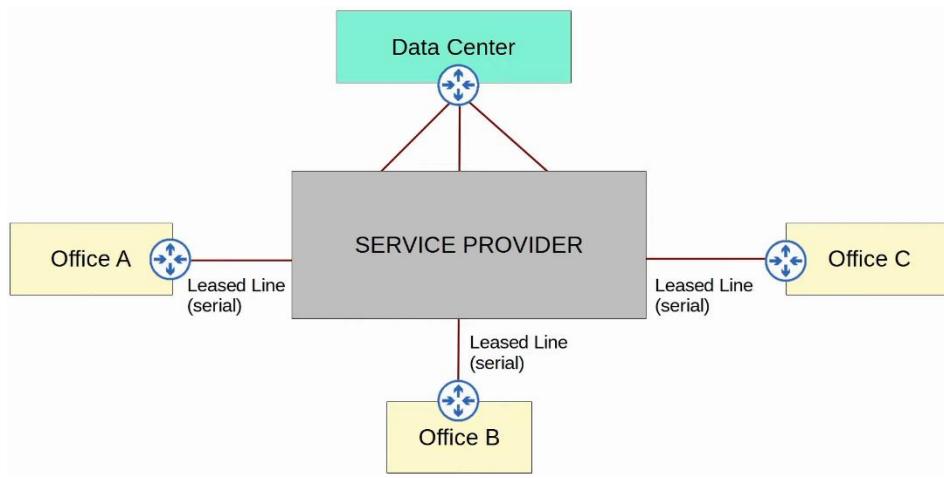
## WAN Architectures

- Wide Area Network
- Extends over a large geographic area
- WANs are used to connect geographically separate LANs
- Although the Internet itself is considered a WAN, the term WAN is usually used to refer to an enterprise's private connections that connect their offices, data centres, and other sites together
- Over public/shared network like Internet, VPNs (Virtual Private Networks) can be used to create private WAN connections
- There have been many different WAN technologies over the years. Depending on the location, some may be available and some may not
- Technologies which are considered 'legacy' (old) in 1 country may still be used in another

### WAN over dedicated connection (Leased Line)

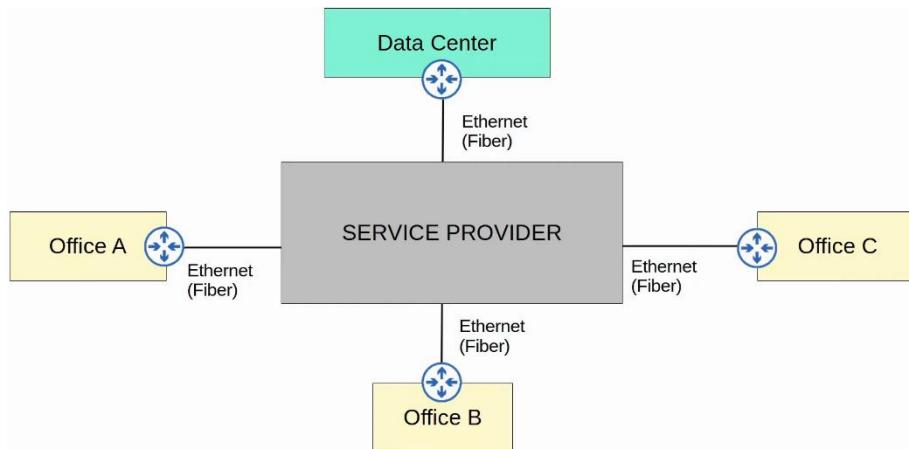


- Hub and Spoke Topology

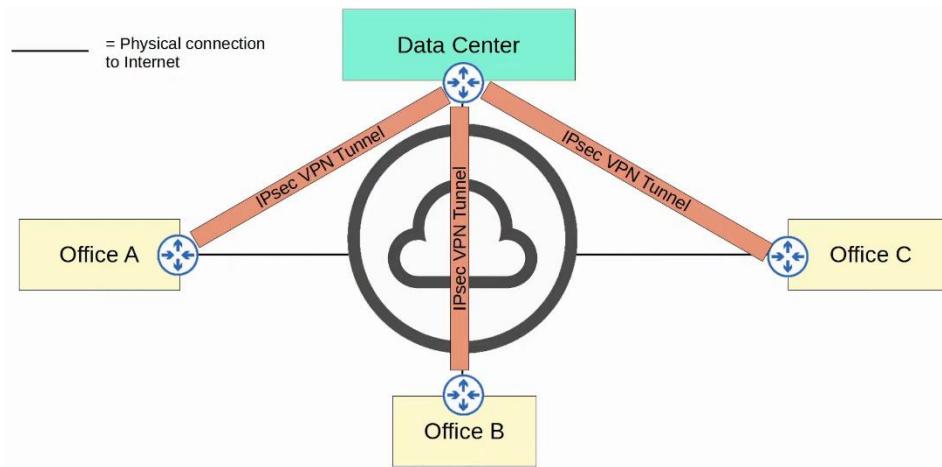


- Better representation of Leased Line

#### **WAN connection via Ethernet (Fiber)**



#### **WAN over shared Infrastructure (Internet VPN)**



## Leased Line

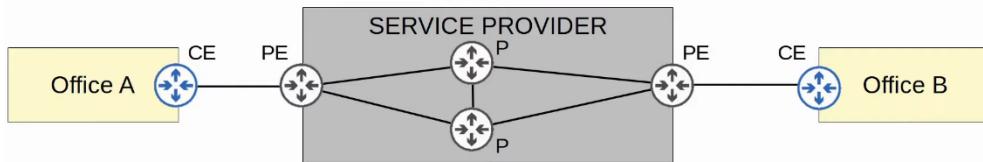
- A dedicated physical link, typically connecting 2 sites
- Leased Lines use serial connections (PPP or HDLC encapsulation)
- There are various standards that provide different speeds, and different standards are available in different countries
- Due to the higher cost, higher installation lead time, and slower speeds of leased lines, Ethernet WAN technologies are becoming more popular

System	North American	Japanese	European (CEPT)
<b>Level zero (channel data rate)</b>	64 kbit/s (DS0)	64 kbit/s	64 kbit/s
<b>First level</b>	1.544 Mbit/s (DS1) (24 user channels) (T1)	1.544 Mbit/s (24 user channels)	2.048 Mbit/s (32 user channels) (E1)
<b>(Intermediate level, T-carrier hierarchy only)</b>	3.152 Mbit/s (DS1C) (48 Ch.)	—	—
<b>Second level</b>	6.312 Mbit/s (DS2) (96 Ch.) (T2)	6.312 Mbit/s (96 Ch.), or 7.786 Mbit/s (120 Ch.)	8.448 Mbit/s (128 Ch.) (E2)
<b>Third level</b>	44.736 Mbit/s (DS3) (672 Ch.) (T3)	32.064 Mbit/s (480 Ch.)	34.368 Mbit/s (512 Ch.) (E3)
<b>Fourth level</b>	274.176 Mbit/s (DS4) (4032 Ch.)	97.728 Mbit/s (1440 Ch.)	139.264 Mbit/s (2048 Ch.) (E4)
<b>Fifth level</b>	400.352 Mbit/s (DS5) (5760 Ch.)	565.148 Mbit/s (8192 Ch.)	565.148 Mbit/s (8192 Ch.) (E5)

## MPLS

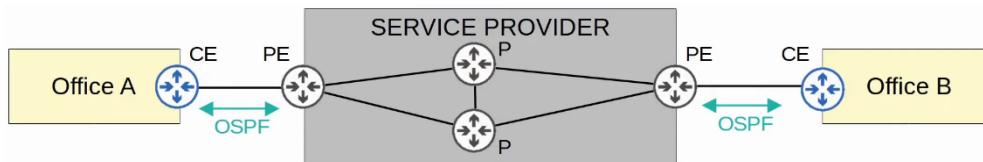
- Multi-Protocol Label Switching
- Similar to the Internet, service providers' MPLS networks are shared infrastructure because many customer enterprises connect to and share the same infrastructure to make WAN connections
- However, the label switching in the name of MLPS allows VPNs to be created over the MLPS infrastructure through the use of labels
- Some important terms
  - Customer Edge (CE) Router
  - Provider Edge (PE) Router
  - Provider core (P) router

- When the PE routers receive frames from the CE routers, they add a label to the frame
- These labels are used to make forwarding decisions within the service provider network, not the destination IP



### Layer 3 MPLS

- The CE routers don't use MPLS, it is only used by PE and P routers
- When using a Layer 3 MPLS VPN, the CE and PE routers peer using OSPF, for example, to share routing information
  - The CE can use the PE as the next hop of their static routes also
- For example, Office A's CE will peer with one PE, and Office B's CE will peer with the other PE
- Office A's CE will learn about Office B's routes via this OSPF peering, and Office B's CE will learn about Office A's routes too

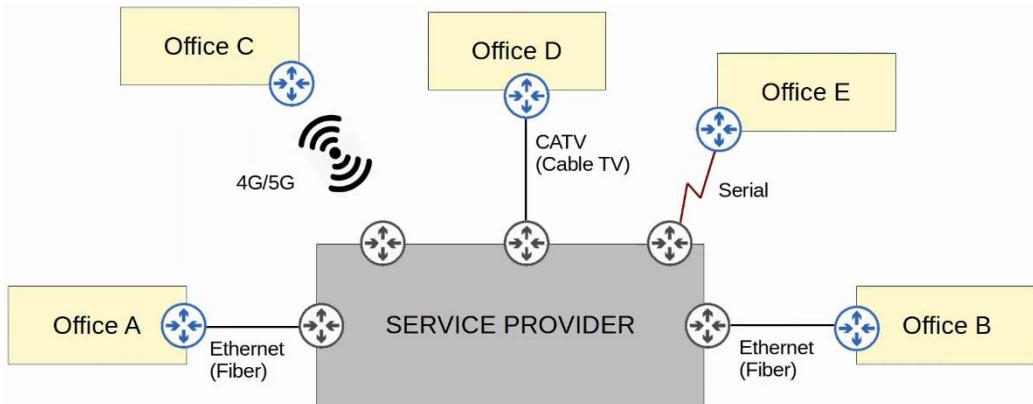


### Layer 2 MPLS

- When using a Layer 2 MPLS VPN, the CE and PE routers do not form peerings
- The service provider network is entirely transparent to the CE routers
- In effect, it is like the 2 CE routers are directly connected
  - Their WAN interfaces will be in the same subnet
- If a routing protocol is used, the 2 CE routers will peer directly with each other
  - The service provider is still running MPLS, but doing so in a way that it is acting like a switch



- Many different technologies can be used to connect to a service provider's MPLS network for WAN service



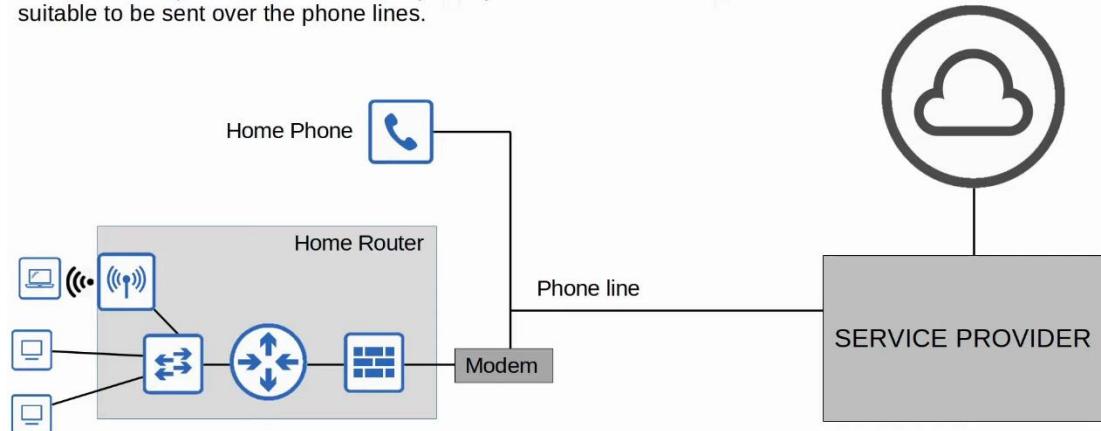
### Internet connections

- There are countless of ways for an enterprise to connect to the Internet
- E.g. private WAN technologies such as leased lines and MPLS VPNs can be used to connect to a service provider's internet infrastructure
- In addition, technologies such as CATV and DSL commonly used by consumers (home Internet access) can also be used by an enterprise
- These days, for both enterprise and consumer internet access , fiber optic Ethernet connections are getting more popular due to high speeds they provide over long distances

### Digital Subscriber Line (DSL)

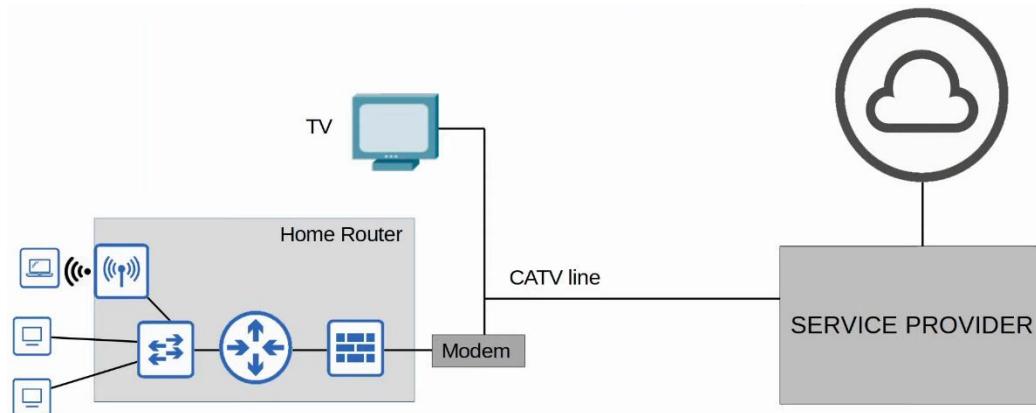
- Provides Internet connectivity to customers over phone lines, and can share the same phone line that is already installed in most homes
- A DSL modem (modulator-demodulator) is required to convert data into a format suitable to be sent over the phone lines
  - The modem might be a separate device or might be incorporated in to the 'home router'

suitable to be sent over the phone lines.



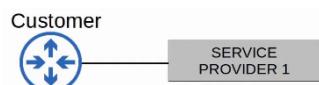
### Cable Internet

- Provides Internet access via the same CATV (Cable Television) lines used for TV service
- Like DSL, a cable modem is required to convert data into a format suitable to be sent over the CATV cables
  - Like a DSL modem, it can be separate device or built into the home router

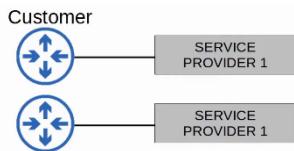


### Redundant Internet Connections

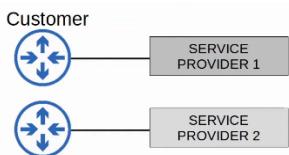
- Single homed
  - 1 connection to 1 ISP



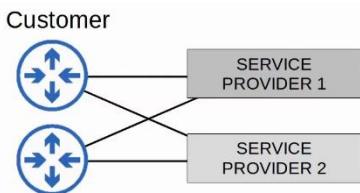
- Dual homed
  - 2 connections to 2 ISP



- Multihomed
  - 1 connection to each of 2 ISPs



- Dual Multihomed
  - 2 connections to each of 2 ISPs

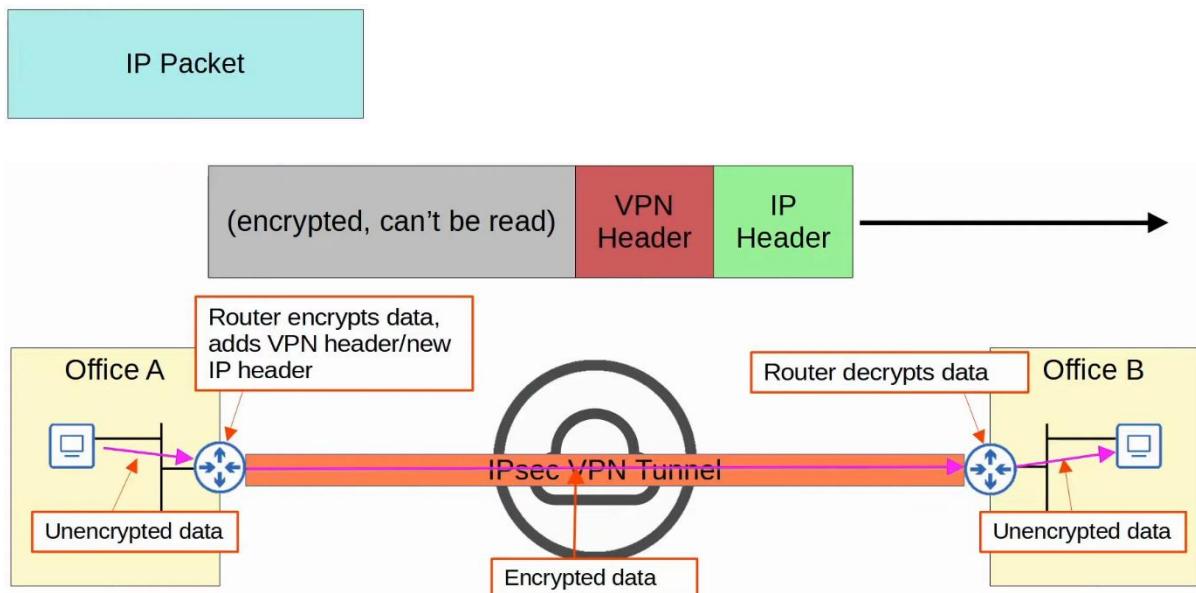


## Internet VPNs

- Private WAN services such as leased lines and MPLS provide security because each customer's traffic is separated by using dedicated physical connections (leased lines) or by MPLS tags
- When using the Internet as a WAN to connect sites together, there is no built-in security by default
- To provide secure communications over the Internet, VPNs are used
- We will cover 2 kinds of Internet VPNs
  - Site-to-Site VPNs using IPSec
  - Remote-accessing VPNs using TLS

## Site-to-Site VPNs (IPSec)

- A VPN btw 2 devices and is used to connect 2 sites together over the Internet
- A VPN 'tunnel' is created btw the 2 devices by encapsulating the original IP packet with a VPN header and new IP header
  - When using IPSec, the original packet is encrypted before being encapsulated with the new header

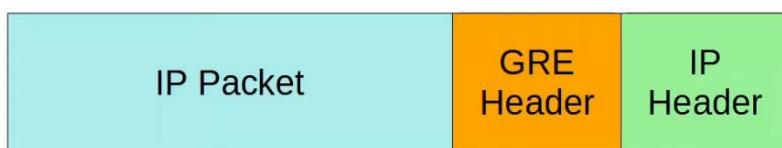


- **Summary**
  1. The sending device combines the original packet and session key (encryption key) and runs them through an encryption formula
  2. The sending device encapsulates the encrypted packet with a VPN header and a new IP header
  3. The sending device sends the new packet to the device on the other side of the tunnel
  4. The receiving device decrypts the data to get the original packet, and then forwards the original packet to its destination
- In a site-to-site VPN, a tunnel is formed only btw 2 tunnel endpoints (e.g. the 2 routers connected to the internet)
- All other devices in each site don't need to create a VPN for themselves. They can send unencrypted data to their site's router, which will encrypt it and forward it in the tunnel as described above
- **Limitations**
  - IPSec don't support broadcast and multicast traffic, only unicast
    - Routing protocols such as OSPF can't be used over the tunnels because they rely on multicast traffic
    - Can be solved with "GRE over IPSec"
  - Configuring a full mesh of tunnels btw many sites is labour-intensive

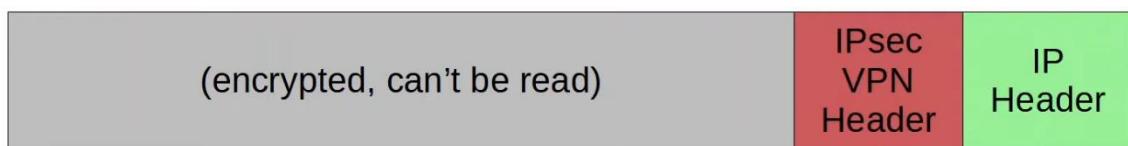
- Can be solved using Cisco's DMVPN

### GRE over IPSec

- GRE (Generic Routing Encapsulation) creates tunnels like IPSec, however, it does not encrypt the original packet, so it is not secure
- However, it has the advantage of being able to encapsulate a wide variety of Layer 3 protocols as well as broadcast and multicast messages
- To get the flexibility of GRE with the security of IPSec, GRE over IPSec can be used
- The original packet will be encapsulated by a GRE header and a new IP header, and then the GRE packet will be encrypted and encapsulated within an IPSec VPN header and new IP header



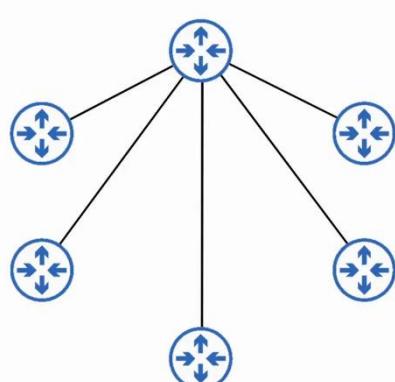
- The above GRE packet is then encrypted as shown below



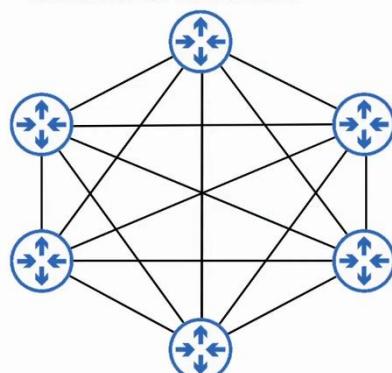
### DMVPN

- DMVPN (Dynamic Multipoint VPN) is a Cisco-developed solution that allows routers to dynamically create a full mesh of IPSec tunnels without having to manually configure every single tunnel
- DMVPN provides the configuration simplicity of hub-and-spoke (each spoke router only needs one tunnel configured) and the efficiency of direct spoke-to-spoke communication (spoke routers can communicate directly without traffic passing through the hub)

1: Configure IPsec tunnels to a hub site.

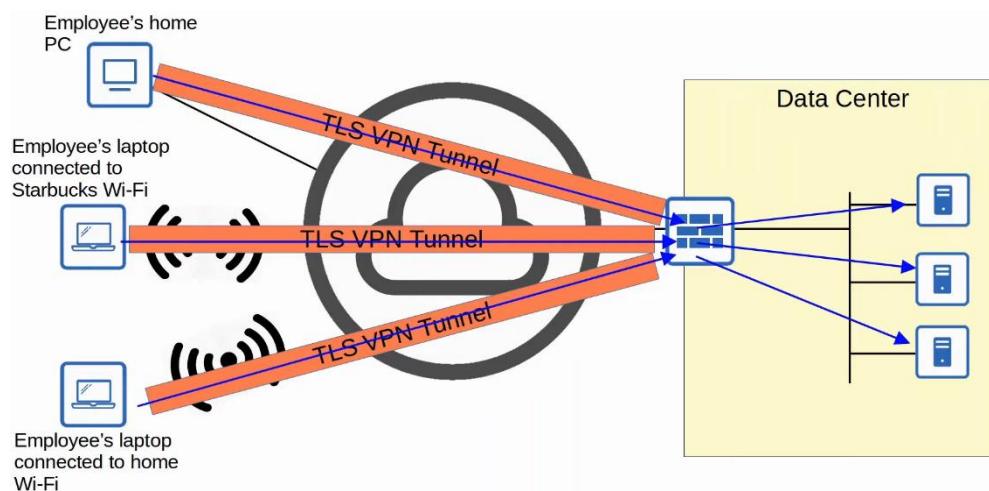


2: The hub router gives each router information about how to form an IPsec tunnel with the other routers.



## Remote-Access VPNs

- Whereas site-to-site VPNs are used to make point-to-point connection btw 2 sites over the Internet, remote-access VPNs are used to allow end devices (PCs, mobile phones) to access the company's internal resources securely over the Internet
- Remote-access VPNs typically use TLS (Transport Layer Security)
  - TLS is also what provides security for HTTPS (HTTP Secure)
  - TLS was formerly known as SSL (Secure Sockets Layer) and developed by Netscape, but it was renamed to TLS when it was standardized by the IETF
- VPN client software (e.g. Cisco AnyConnect) is installed on end devices (e.g. company-provided laptops used by employees to work from home)
- These end devices form secure tunnels to one of the company's routers/firewalls acting as a TLS server
- This allows the end users to securely access resources on the company's internal network without being directly connected to the company network



- TLS also encrypt and add additional headers to packets

## Site-to-Site VS Remote-Access VPN

- SS VPNs typically use IPSec
  - RA VPNs typically use TLS
- 
- SS VPNs provide service to many devices within the sites they are connecting
  - RA VPNs provide service to the one end device the VPN client software is installed on

- SS VPNs are typically used to permanently connect 2 sites over the Internet
- RA VPNs are typically used to provide on-demand access for end devices that want to securely access company resources while connected to a network which is not secure

## Virtualization & Cloud

Things covered

- Intro to Virtualization
  - Virtual servers
  - Virtual networks
- Intro to cloud computing
  - Essential characteristics
  - Service Models
  - Deployment Models
- Connecting to public clouds

### **Server Hardware**

- Although Cisco is more known for their networking devices, they also have hardware servers such as UCS (Unified Computing Systems)
- The larger vendors of hardware servers include DELL EMC, HPE etc

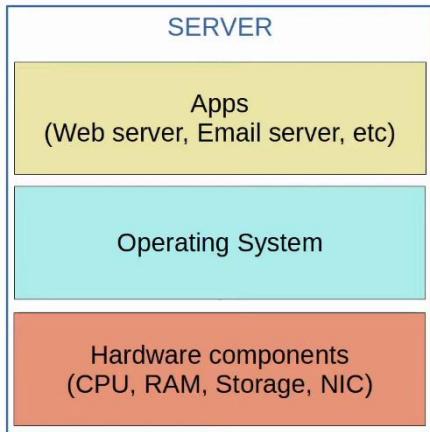


### **Virtualization**

#### **Servers before Virtualization**

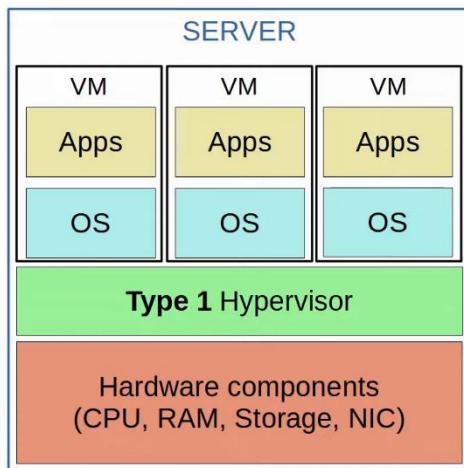
- Before virtualization, there was a 1-to-1 relationship btw a physical server and an OS
- In that OS, apps providing services such as web server, email server, etc would run
- 1 physical server would be used for each service
  - E.g. 1 physical server for email, 1 for web server, etc

- Prevents a problem from a single app from affecting all other apps
- This is inefficient
  - Each physical server is expensive and takes up space, power, etc
  - The resources on each physical server (CPU, RAM, Storage, NIC) are typically under-used



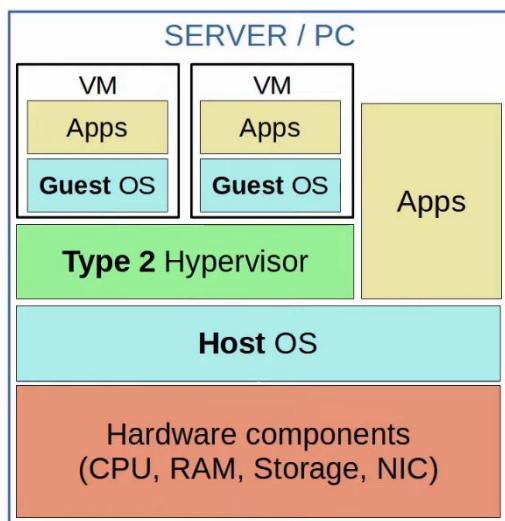
### Type 1 Hypervisor

- Virtualization allows us to break the 1-to-1 relationship of hardware to OS, allowing multiple OS's to run on a single physical server
- Each instance is called a VM (Virtual Machine)
- A hypervisor is used to manage and allocate the hardware resources (CPU, RAM, etc) to each VM
- Another name for a hypervisor is VMM (Virtual Machine Monitor)
- The type of hypervisor which runs directly on top of the hardware is called a Type 1 hypervisor
  - E.g. VMware ESXi, Microsoft Hyper-V, etc
- Type 1 hypervisor are also called bare-metal hypervisors because they run directly on hardware (metal)
  - Another term is native hypervisor
- This is the type used in data center environments



### Type 2 Hypervisor

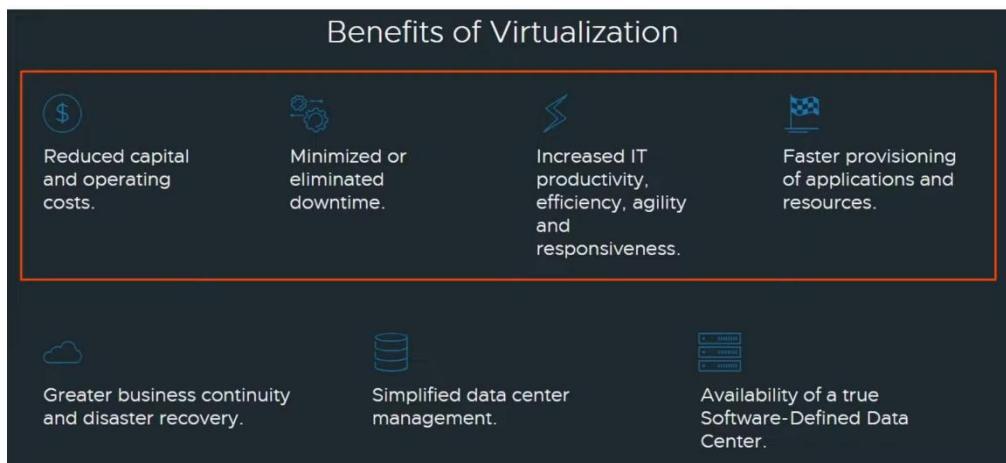
- Type 2 hypervisors run as a program on an OS like a regular computer program
  - E.g. VMware workstation, Oracle VirtualBox, etc
- The OS running directly on the hardware is called the Host OS, and the OS running in a VM is called a Guest OS
- Another name is hosted hypervisor
- Although Type 2 hypervisors are rarely used in data center environments, they are common on personal-use devices
  - E.g. a Mac/Linux user needs to run an app that is only available in Windows



### Why Virtualization?

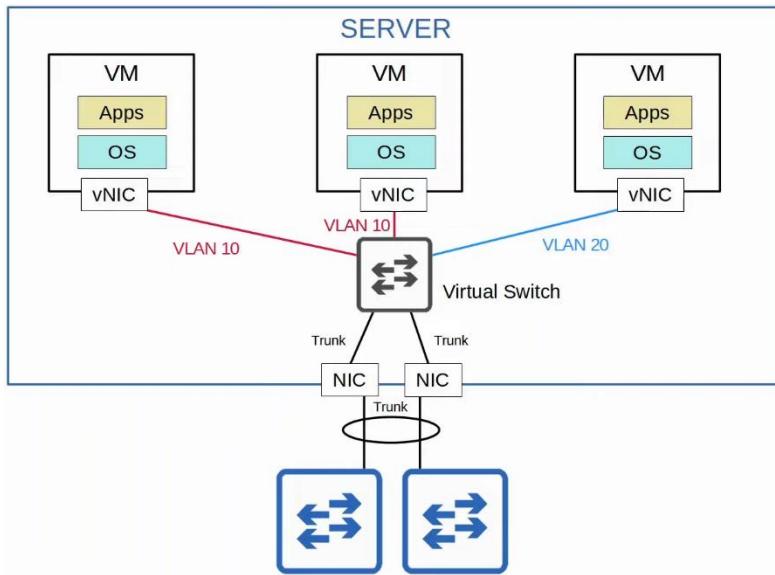
- Partitioning
  - Run multiple OS on 1 physical machine

- Divide system resources btw VMs
- Isolation
  - Provide fault and security isolation at the hardware level
  - Preserve performance with advanced resource controls
- Encapsulation
  - Save the entire state of a VM to files
  - Move and copy VMs as easily as moving and copying files
- Hardware Independence
  - Provision or migrate any VM to any physical server



## Connecting VMs to the Network

- VMs are connected to each other and the external network via a virtual switch running on the hypervisor
- Just like a regular physical switch, the vSwitch's interfaces can operate as access/trunk ports and use VLANs to separate VLANs at Layer 2
- Interfaces on the vSwitch connect to the physical NIC (or NICs) of the server to communicate with the external network
- Virtual Port Channel (VPC) can be formed to connect the NICs with the switches for redundancy



## Cloud Services

- Traditional IT infrastructure deployments were some of the following combination
  - On-Premise
    - All servers, network devices, and other infrastructure are located on company property
    - All equipment is purchased and owned by the company
    - The company is responsible for necessary power, space and cooling
  - Colocation
    - Data centers that rent out space for customers to put their infrastructure (e.g. servers etc)
    - The data centers provides the space, power and cooling
    - The servers are still the responsibility of end customer, although they are not located on the customer's premise
- Cloud services provide an alternative that is hugely popular, and is continuing to grow
- Most people associate 'cloud' with public cloud providers such as AWS
  - Although it is the most common use of cloud services, it's not the only one
  
  
  
  
  
  
  
  
  
- The American NIST (National Institute of Standards and Technology) defined cloud computing in SP (Special Publication) 800-145
- "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with

minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models."

- The main things outlined
  - 5 essential characteristics
  - 3 service models
  - 4 deployment models

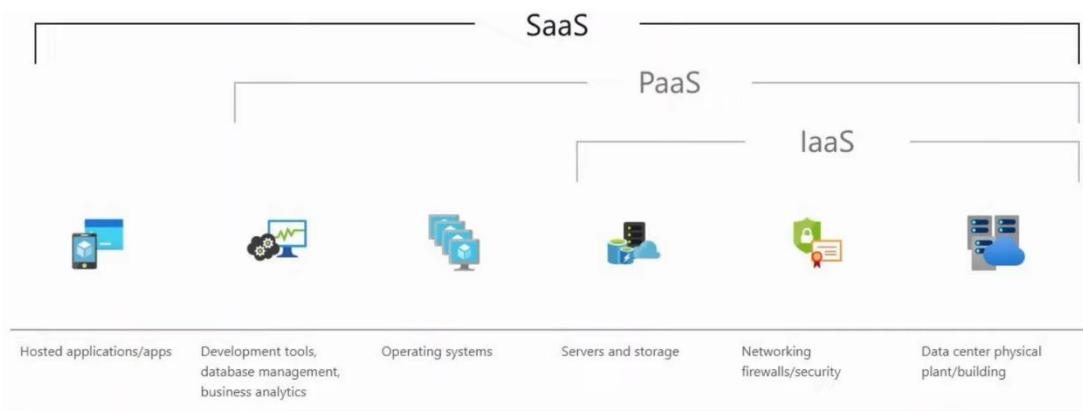
## 5 Essential Characteristics

- **On-demand self-service**
  - "A customer can unilaterally provision computing capabilities, such as server time, and network storage, as needed automatically w/o requiring human interaction with each service provider"
  - The customer is able to use the service (or stop the service) freely (via web portal) w/o direct communication with the service provider
- **Broad network access**
  - "Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g. mobile phones, tablets, laptops, and workstations)
  - The service is available through standard network connections (e.g. Internet, private WAN connections) and can be accessed through many kind of devices
- **Resource pooling**
  - A pool of resources is provided by the service provider, and when a customer requests a service (e.g. creates a new VM), the resources to fulfil that request are allocated from the shared pool
- **Rapid elasticity**
  - Customers can quickly expand the services they use in the cloud (e.g. add new VMs, expand storage, etc) from a pool of resources that appears to be infinite
  - Likewise, they can quickly reduce their services when not needed
- **Measured service**
  - The cloud service provider measures the customer's usage of cloud resources, and the customer can measure their own use as well
  - Customers are charged based on usage

## 3 Service Models

- In cloud computing, everything is provided on a 'service' model
- E.g. Rather than buying a physical server, mounting it on a rack, installing the hypervisor, creating VMs, etc, the service provider offers all of this as a service
- There are a variety of services referred to as '\_ as a service' or '\_aaS'
- The 3 service models are
  - **Software as a Service (SaaS)**
    - The capability provided to the consumer is to use the provider's applications running on cloud services
    - Microsoft Office 365 is a popular example of SaaS

- **Platform as a Service (PaaS)**
  - The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications using programming languages, libraries, services, and tools supported by the provider
  - Service provider provides a platform for developers to run applications
  - E.g. AWS Lambda, Google App Engine
- **Infrastructure as a Service (IaaS)**
  - Capability provided is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include OS and applications
  - Consumer cannot control underlying cloud infrastructure, but can control OS, storage and deployed applications
  - E.g. EC2 and Google Compute Engine



## 4 Deployment Models

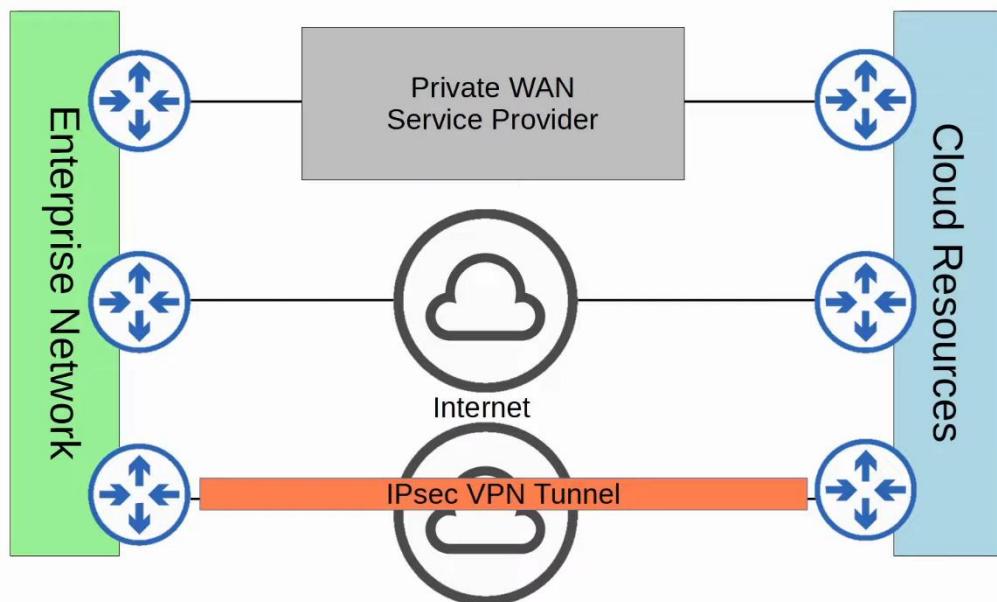
- Most people assume 'cloud' means public providers such as AWS, Azure, and GCP
- Although Public cloud is the most common deployment model, it's not the only one
- 4 deployment models
  - **Private cloud**
    - Generally used by large enterprises only
    - Although private, may be owned by a 3rd party
      - E.g. AWS provides private cloud services to American DoD
    - May be on/off premises
    - Same kind of services as in public clouds are offered (SaaS, PaaS, IaaS), but the infrastructure is reserved for a single organization
  - **Community cloud**

- Least common
- Same as public, but a group instead of 1 organization
- **Public cloud**
  - Most common
  - Can be used by the public
- **Hybrid cloud**
  - A combination of the 3 deployment types
  - E.g. a private cloud which can offload to a public cloud when necessary

## **Benefits of Cloud**

- **Cost**
  - Capital Expenses of buying hardware, setting up data centers, etc are reduced or eliminated
- **Global scale**
  - Cloud services can scale globally at a rapid pace
  - Services can be set up and offered to customers from a geographic location close to them
- **Speed/Agility**
  - Services are provided on demand, and vast amounts of resources can be provisioned within minutes
- **Productivity**
  - Cloud services remove the need for many time-consuming tasks such as procuring physical servers, racking them, cabling, installing and updating OS, etc
- **Reliability**
  - Backups in the cloud are very easy to perform
  - Data can be mirrored at multiple sites in different geographic locations to support disaster recovery

## **Connecting to Cloud Resources**



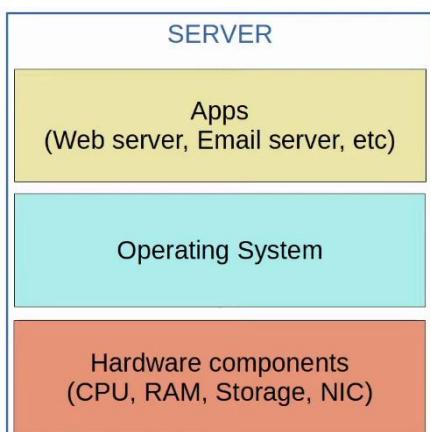
Virtualization: Containers

Things covered

- Review of VMs
- Containers
- VM vs Container

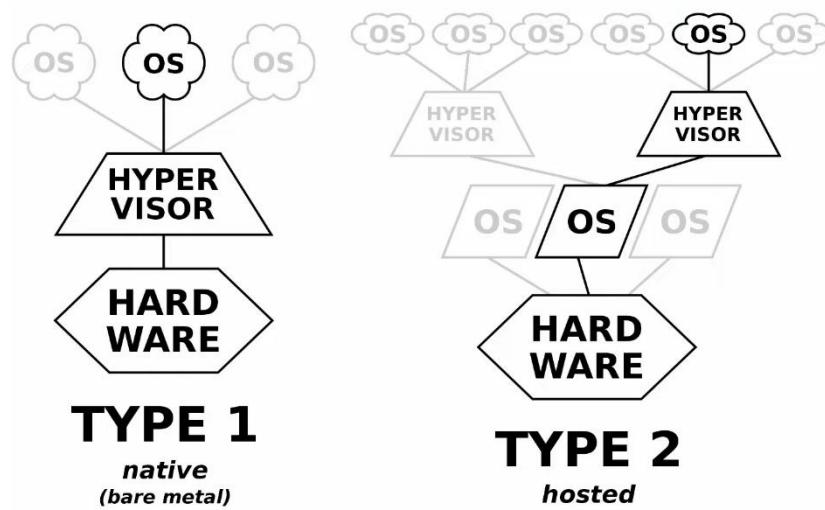
## Virtual Machines

W/o Virtualization



## With Virtualization

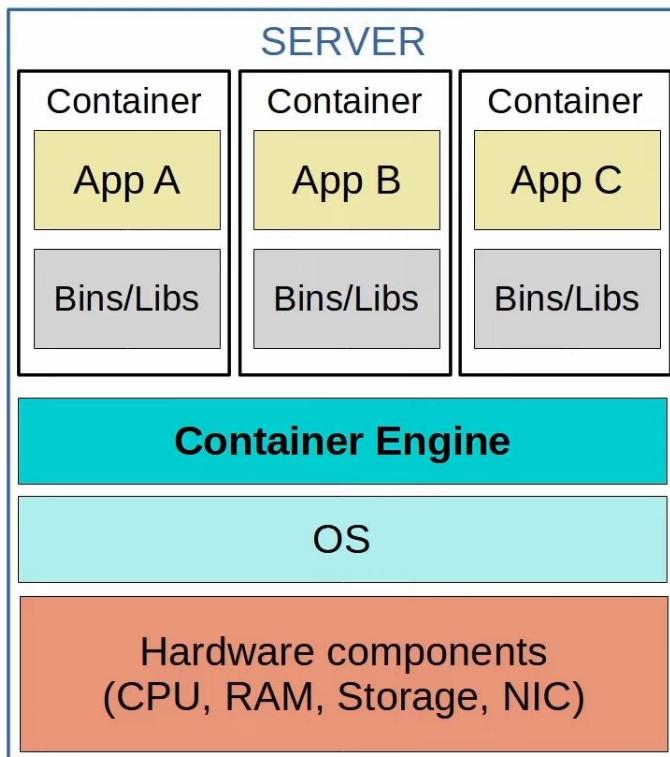
- VMs allow multiple OS's to run on a single physical server.
- A Hypervisor is used to manage and allocate hardware resources to each VM.
  - Type 1 Hypervisors (aka Native or Bare-metal) run directly on top of hardware.
  - Type 2 Hypervisors (aka Hosted) run on top of a Host OS (ie. Windows).
- Bins/Libs are the software libraries/services needed by the Apps running in each VM.
- A VM allows its app/apps to run in an isolated environment, separate from the apps in other VMs.
- VMS are easy to create, delete, move, etc.
  - A VM can be easily saved and moved between different physical servers.



## Containers

- Containers are software packages that contain an App and all dependencies (Bins/Libs in the diagram) for the contained App to run.
  - Multiple Apps can be run in a single container, but this is not how containers are usually used
- Containers run on a Container Engine (i.e. Docker Engine)
  - The container engine is run on a host OS (usually Linux)
- Containers are lightweight (small in size) and include only the dependencies required to run the specific App
- A Container Orchestrator is a software platform for automating the deployment, management, scaling etc. of containers

- Kubernetes (originally designed by Google) is the most popular container orchestrator
  - Docker Swarm is Docker's container orchestration tool.
- In small numbers manual operation is possible, but large-scale systems (ie. with Microservices) can require thousands of containers
- Microservice Architecture is an approach to software architecture that divides a larger solution into smaller parts (microservices).
  - Those microservices all run in containers that can be orchestrated by Kubernetes (or another platform).



## VM vs Container

- Boot up time
  - VM: take minutes as each VM runs its own OS
  - Containers: in milliseconds
- Space
  - VM: use more disk space (GB)
  - Container: very little disk space (MB)
- Resources
  - VM: use more CPU/RAM resources (each VM runs its own OS)
  - Container: use fewer CPU/RAM (shared OS)
- Portability
  - VM: portable and can move between physical systems running the same hypervisor
  - Container: more portable, they are smaller, faster to boot up, and Docker containers can be run on nearly any container service
- Reliability

- VM: more isolated as each VM runs its own OS
- Container: less isolated because they all run on the same OS; if the OS crash, all containers running on it are affected

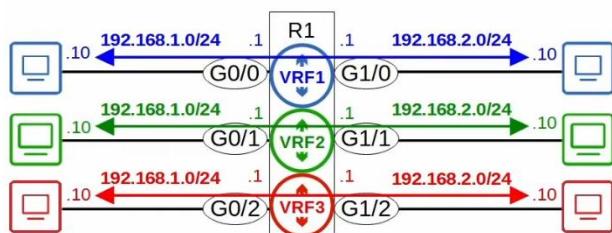
## Virtual Routing & Forwarding (VRF)

Things covered

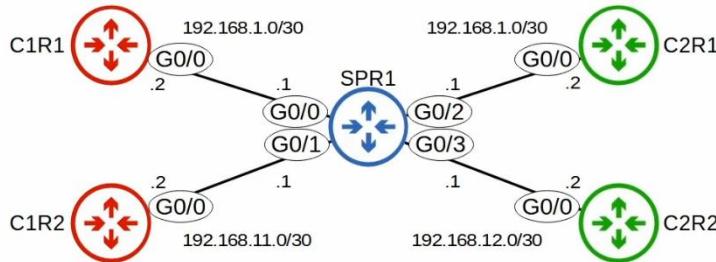
- Intro to VRF
- Config

### VRF (Virtual Routing & Forwarding)

- VRF is used to divide a single router into multiple virtual routers
  - Similar to how VLANs are used to divide a single switch (LAN) into multiple virtual switches (VLAN)
- Does this by allowing a router to build multiple separate routing tables
  - Interfaces (Layer 3 only) and routes are configured to be in specific VRF (aka VRF instance)
  - Router interfaces, SVIs and routed ports on multilayer switches can be configured in a VRF
- Traffic in 1 VRF cannot be forwarded out of an interface in another VRF
  - As an exception, VRF leaking can be configured to allow traffic to pass btw VRFs
- VRF is commonly used to facilitate MPLS
  - The kind of VRF we are talking about is VRF-lite (VRF w/o MPLS)
- VRF is commonly used by service providers to allow 1 device to carry traffic from multiple customers
  - Each customer's traffic is isolated from the others
  - Customer IP addresses can overlap w/o issues



### Config



```

SPR1(config)# interface g0/0
SPR1(config-if)# ip address 192.168.1.1 255.255.255.252
SPR1(config-if)# no shutdown

SPR1(config-if)# interface g0/1
SPR1(config-if)# ip address 192.168.11.1 255.255.255.252
SPR1(config-if)# no shutdown

SPR1(config-if)# interface g0/2
SPR1(config-if)# ip address 192.168.1.1 255.255.255.252
% 192.168.1.0 overlaps with GigabitEthernet0/0

SPR1(config-if)# ip address 192.168.1.2 255.255.255.252
% 192.168.1.0 overlaps with GigabitEthernet0/0

```

G0/2 cannot use IP address 192.168.1.1 because it is in the same subnet as G0/0 (in this case it's the exact same IP address).

Even if the IP address is different, G0/2 cannot be configured in the same subnet as G0/0.

Without the use of VRF, two interfaces on the same router cannot be in the same subnet.

```

SPR1(config)# ip vrf CUSTOMER1
SPR1(config-vrf)# ip vrf CUSTOMER2
SPR1(config-vrf)# do show ip vrf
  Name           Default RD      Interfaces
  CUSTOMER1      <not set>
  CUSTOMER2      <not set>

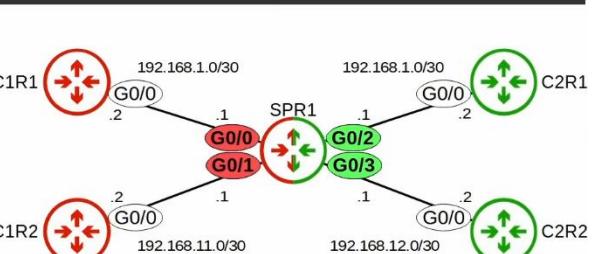
SPR1(config-vrf)# interface g0/0
SPR1(config-if)# ip vrf forwarding CUSTOMER1
% Interface GigabitEthernet0/0 IPv4 disabled and address(es) removed due to enabling VRF CUSTOMER1
SPR1(config-if)# ip address 192.168.1.1 255.255.255.252

SPR1(config-if)# interface g0/1
SPR1(config-if)# ip vrf forwarding CUSTOMER1
% Interface GigabitEthernet0/1 IPv4 disabled and address(es) removed due to enabling VRF CUSTOMER1
SPR1(config-if)# ip address 192.168.11.1 255.255.255.252

SPR1(config-if)# interface g0/2
SPR1(config-if)# ip vrf forwarding CUSTOMER2
SPR1(config-if)# ip address 192.168.12.1 255.255.255.252
SPR1(config-if)# no shutdown
SPR1(config-if)# do show ip vrf
  Name           Default RD      Interfaces
  CUSTOMER1      <not set>
  CUSTOMER2      <not set>

```

1. Create VRFs:  
SPR1(config)# ip vrf name  
2. Assign interfaces to VRFs:  
SPR1(config-if)# ip vrf forwarding name



```

SPR1# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route, H - NHRP, L - LISP
      a - application route
      + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

```

**show ip route** displays the global routing table.  
\*All of SPR1's interfaces are configured in VRFs, so nothing displays here.  
\*You can have a mix of interfaces using and not using VRFs.

```

SPR1# show ip route vrf CUSTOMER1

Routing Table: CUSTOMER1
!output omitted

      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/30 is directly connected, GigabitEthernet0/0
L        192.168.1.1/32 is directly connected, GigabitEthernet0/0
      192.168.11.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.11.0/30 is directly connected, GigabitEthernet0/1
L        192.168.11.1/32 is directly connected, GigabitEthernet0/1

SPR1# show ip route vrf CUSTOMER2

Routing Table: CUSTOMER2
!output omitted

      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/30 is directly connected, GigabitEthernet0/2
L        192.168.1.1/32 is directly connected, GigabitEthernet0/2
      192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.12.0/30 is directly connected, GigabitEthernet0/3
L        192.168.12.1/32 is directly connected, GigabitEthernet0/3

```

```

SPR1# ping 192.168.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

SPR1# ping vrf CUSTOMER1 192.168.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

SPR1# ping vrf CUSTOMER1 192.168.11.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.11.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

SPR1# ping vrf CUSTOMER1 192.168.12.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

SPR1# ping vrf CUSTOMER2 192.168.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

SPR1# ping vrf CUSTOMER2 192.168.12.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

```

# Wireless

## Wireless Fundamentals

Things covered

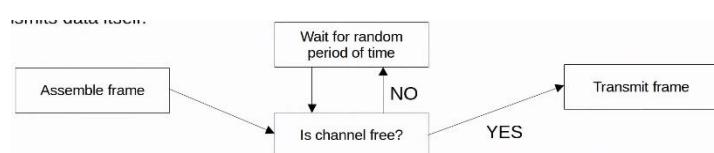
- Radio Frequency (RF)
- Wi-fi standards
- Wireless LAN Fundamentals

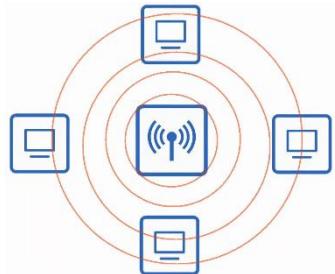
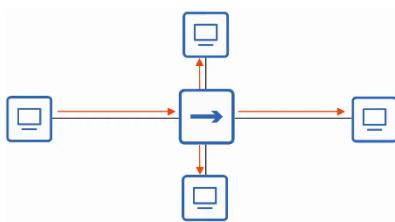
## Wireless Networks

- The standards used for wireless LAN is defined in IEEE 802.11
- The term 'Wi-Fi' is a trademark of the 'Wi-Fi Alliance', not directly connected to IEEE
  - They test and certify equipment for 802.11 standards compliance interoperability with other devices
  - However, wifi has become the common term to refer to 802.11 wireless LAN

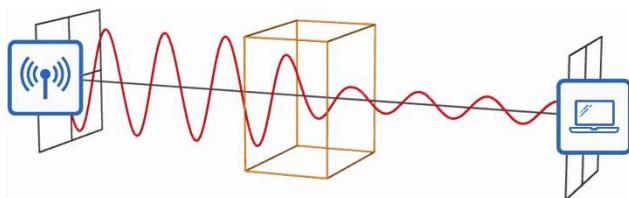


- Wireless networks have some issues that needs to be dealt with
  1. All devices within range receive all frames, like devices connected to an Ethernet hub
    - Privacy of data within the LAN is a greater concern
    - CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) is used to facilitate half duplex communications
      - CSMA/CD used in wired networks to detect and recover from collisions
      - CSMA/CA used in wireless networks to avoid collisions
      - When using CSMA/CA , a device will wait for other devices to stop transmitting before it transmits data itself

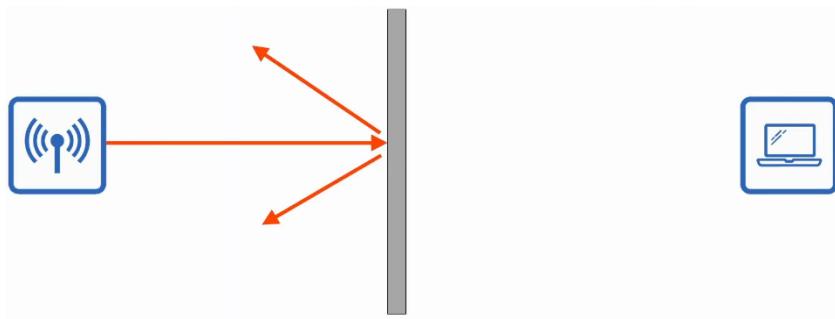




2. Wireless communications are regulated by various international and national bodies
  - Covered in IEEE 802.11
  
  
  
3. Wireless signal coverage area must be considered
  - Signal range
  - Signal absorption, reflection, refraction, diffraction, and scattering
    - Signal Absorption
      - Happens when a wireless signal passes through a material and is converted into heat, weakening the original signal

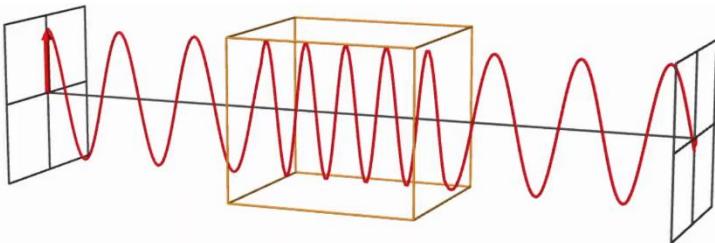


- Signal Reflection
  - Happens when a signal bounces off a material, e.g. metal
  - This is why Wi-Fi reception is usually poor in elevators, the signal bounce off the metal and very little penetrates into the elevator



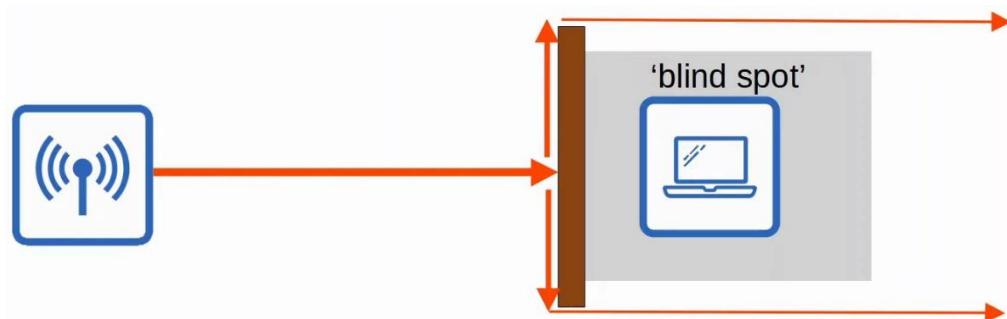
- Signal Reflection

- Happens when a wave is bent when entering a medium where the signal travels in a different speed
- E.g. Glass and water can refract waves



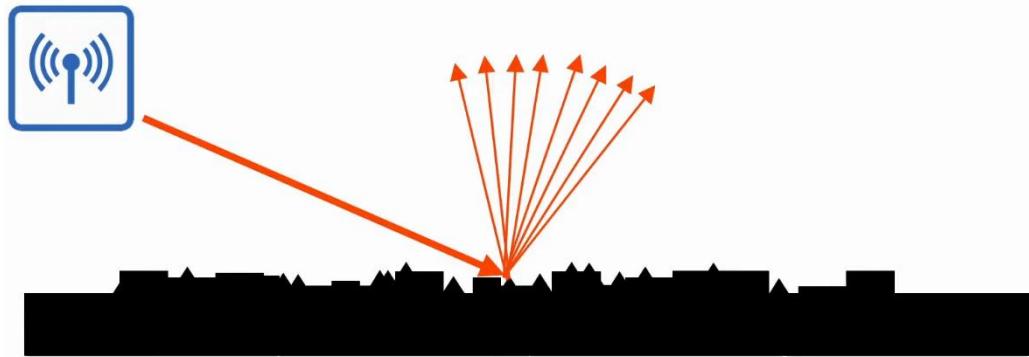
- Signal Diffraction

- Happens when a wave encounters an obstacle and travels around it
- This can result in blind spots behind the obstacle



- Signal Scattering

- Happens when a material causes a signal to scatter in all directions
- E.g. dust, smog, uneven surfaces can cause scattering



4. Other devices using the same channels can cause interference
- E.g. a wireless LAN in your neighbour's house/apartment

### Radio Frequency (RF)

- To send wireless signals, the sender applies an alternating current to an antenna
  - This creates EM fields which propagate out as waves
- EM waves can be measured in multiple ways, e.g. amplitude and frequency
  - Amplitude is the max strength of the electric and magnetic fields
  - Frequency measures the number of up/down cycles per unit time
    - Hertz (Hz)
    - Period = time of 1 cycle
- Visible frequency: 400Thz - 790Thz
- Radio frequency: 30Hz - 300GHz

Band name	Abbreviation	ITU band number	Frequency and Wavelength	Example Uses
Extremely low frequency	ELF	1	3–30 Hz 100,000–10,000 km	Communication with submarines
Super low frequency	SLF	2	30–300 Hz 10,000–1,000 km	Communication with submarines
Ultra low frequency	ULF	3	300–3,000 Hz 1,000–100 km	Submarine communication, communication within mines
Very low frequency	VLF	4	3–30 kHz 100–10 km	Navigation, time signals, submarine communication, wireless heart rate monitors, geophysics
Low frequency	LF	5	30–300 kHz 10–1 km	Navigation, time signals, AM longwave broadcasting (Europe and parts of Asia), RFID, amateur radio
Medium frequency	MF	6	300–3,000 kHz 1,000–100 m	AM (medium-wave) broadcasts, amateur radio, avalanche beacons
High frequency	HF	7	3–30 MHz 100–10 m	Shortwave broadcasts, citizens band radio, amateur radio and over-the-horizon aviation communications, RFID, over-the-horizon radar, automatic link establishment (ALE) / near-vertical incidence skywave (NVIS) radio communications, marine and mobile radio telephony
Very high frequency	VHF	8	30–300 MHz 10–1 m	FM, television broadcasts, line-of-sight ground-to-aircraft and aircraft-to-aircraft communications, land mobile and maritime mobile communications, amateur radio, weather radio
Ultra high frequency	UHF	9	300–3,000 MHz 10–1 mm	Television broadcasts, microwave oven, microwave devices/communications, radio astronomy, mobile phones, wireless LAN, Bluetooth, ZigBee, GPS and two-way radios such as land mobile, FRS and GMRS radios, amateur radio, satellite radio, Remote control Systems, ADSB
Super high frequency	SHF	10	3–30 GHz 100–10 mm	Radio astronomy, microwave devices/communications, wireless LAN, DSRC, most modern radars, communications satellites, cable and satellite television broadcasting, DBS, amateur radio, satellite radio
Extremely high frequency	EHF	11	30–300 GHz 10–1 mm	Radio astronomy, high-frequency microwave radio relay, microwave remote sensing, amateur radio, directed-energy weapon, millimeter wave scanner, Wireless Lan 802.11ad
Terahertz or Tremendously high frequency	THz or THF	12	300–3,000 GHz 1–0.1 mm	Experimental medical imaging to replace X-rays, ultrafast molecular dynamics, condensed-matter physics, terahertz time-domain spectroscopy, terahertz computing/communications, remote sensing

### Radio Frequency Bands

- Wi-Fi uses 2 main bands (frequency ranges)
  - 2.4GHz band
    - 2.400GHz - 2.4835GHz
  - 5GHz band

- 5.150GHz - 5.825GHz
- Divided into 4

5.150 - 5.	250
------------	-----

- 5.250 - 5.350
- 5.470 - 5.725
- 5.725 - 5.825

- The 2.4GHz band typically provides further reach in open space and better penetration of obstacles such as walls
  - However, more devices tend to use this band so interference can be a bigger problem compared to 5GHz
- Wi-Fi 6 (802.11ax) has expanded the spectrum range to include a band in the 6GHz range

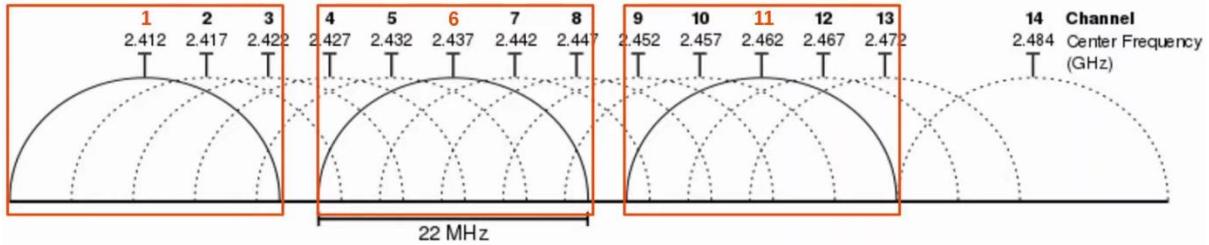
## Channels

- Each band is divided up into multiple channels
  - Devices are configured to transmit and receive traffic on 1 (or more) of these channels
- The 2.4GHz band is divided into several channels, each with a 22MHz range

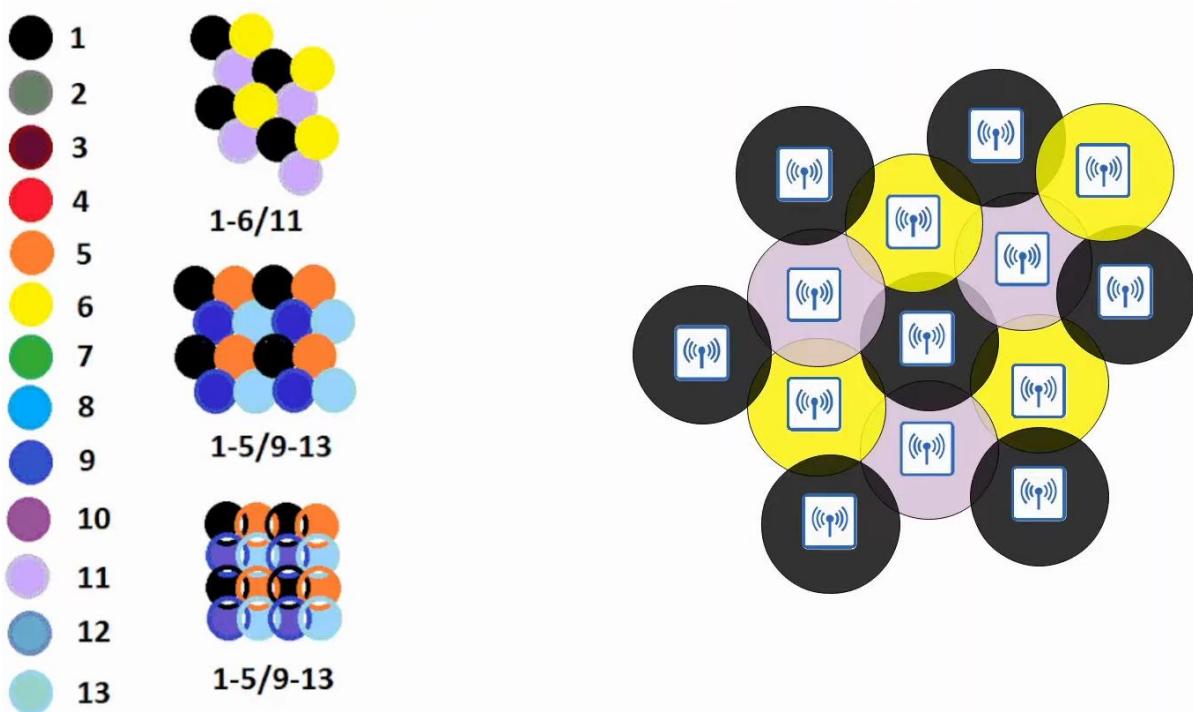
Channel	F <sub>0</sub> (MHz)	Frequency Range (MHz)	North America <sup>[8]</sup>	Japan <sup>[8]</sup>	Most of world <sup>[8][9][10][11][12][13][14][15]</sup>
1	2412	2401–2423	Yes	Yes	Yes
2	2417	2406–2428	Yes	Yes	Yes
3	2422	2411–2433	Yes	Yes	Yes
4	2427	2416–2438	Yes	Yes	Yes
5	2432	2421–2443	Yes	Yes	Yes
6	2437	2426–2448	Yes	Yes	Yes
7	2442	2431–2453	Yes	Yes	Yes
8	2447	2436–2458	Yes	Yes	Yes
9	2452	2441–2463	Yes	Yes	Yes
10	2457	2446–2468	Yes	Yes	Yes
11	2462	2451–2473	Yes	Yes	Yes
12	2467	2456–2478	No <sup>B</sup> except CAN	Yes	Yes
13	2472	2461–2483	No <sup>B</sup>	Yes	Yes
14	2484	2473–2495	No	11b only <sup>C</sup>	No

- 11b in Japan is an old technology
- Channels 1- 5 overlap each other, so if wireless points are next to each other, need to be careful which channels to choose

- In a small wireless LAN with only a single AP, can use any channel
- However, in larger WLANs with multiple APs, it's important that adjacent APs don't use overlapping channels
  - Help avoid interference
- In the 2.4GHz band, it is recommended to use channels 1, 6 and 11



- Outside of North America, can use other combinations, but for CCNA - 1, 6, 11
- The 5GHz band consists of non-overlapping channels, so it is much easier to avoid interference btw adjacent APs
- Using channels 1, 6 and 11, you can place APs in a honeycomb pattern to provide a complete coverage of an area w/o interference btw channels



- The pattern on the left include combinations from the rest of the world

## 802.11 Standards

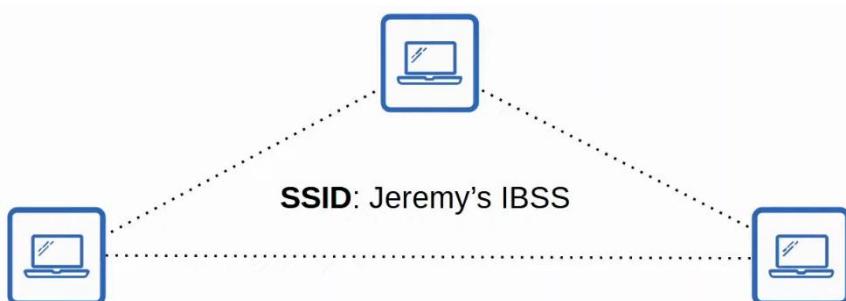
Standard	Frequencies	Max Data Rate (theoretical)	Alternate Name
802.11	2.4 GHz	2 Mbps	
802.11b	2.4 GHz	11 Mbps	
802.11a	5 GHz	54 Mbps	
802.11g	2.4 GHz	54 Mbps	
802.11n	2.4 / 5 GHz	600 Mbps	'Wi-Fi 4'
802.11ac	5 GHz	6.93 Gbps	'Wi-Fi 5'
802.11ax	2.4 / 5 / 6 GHz	4*802.11ac	Wi-Fi 6'

## Service Sets

- 802.11 defines different kinds of service sets which are groups of wireless network devices
- There are 3 main types
  - Independent
  - Infrastructure
  - Mesh
- All devices in a service set share the same SSID (service set identifier)
- The SSID is a human-readable name which identifies the service set
- The SSID does not have to be unique

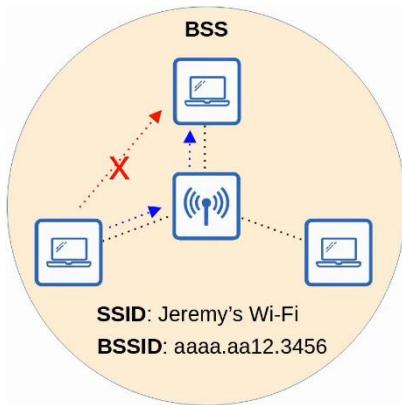
## IBSS (Independent Basic Service Set)

- A wireless network in which 2 or more wireless device connect directly without using an AP (Access Point)
- Also called 'ad hoc network'
- Can be used for file transfer (e.g. AirDrop)
- Not scalable beyond a few devices



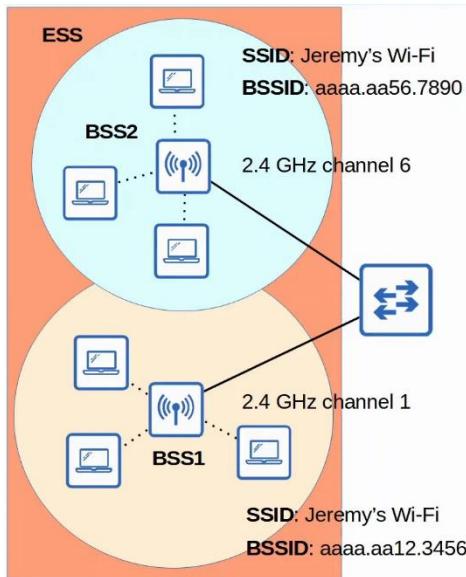
## BSS (Basic Service Set)

- It is a kind of Infrastructure Service Set in which clients connect to each other via an AP (access point), but not directly to each other
- A BSSID (BSS ID) is used to uniquely identify the AP
  - Other APs can use the same SSID, but not the same BSSID
  - The BSSID is the MAC address of the AP's radio
- Wireless devices request to associate with the BSS
- Wireless devices that have associated with the BSS are called 'clients' or 'stations'
- \*Clients must communicate via the AP and not with each other
- \*The area around an AP where its signal is usable is called a BSA (Basic Service Area)
- BSS is a group of devices
- BSA is an area



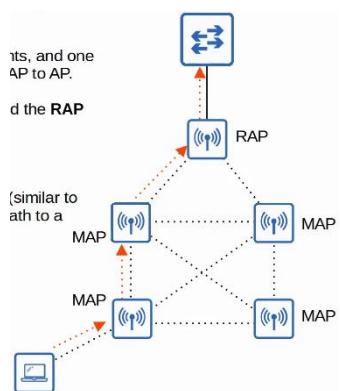
## ESS (Extended Service Set)

- To create larger wireless LANs beyond the range of a single AP, use an ESS
- APs with their own BSS are connected by a wired network
  - Each BSS uses the same SSID
  - Each BSS has a unique BSSID
  - Each BSS uses a different channel to avoid interference
- Clients can pass between APs w/o having to reconnect, providing a seamless Wi-Fi experience when moving between APs
  - Called 'roaming'
- The BSA should overlap 10-15%



### MBSS (Mesh Basic Service Set)

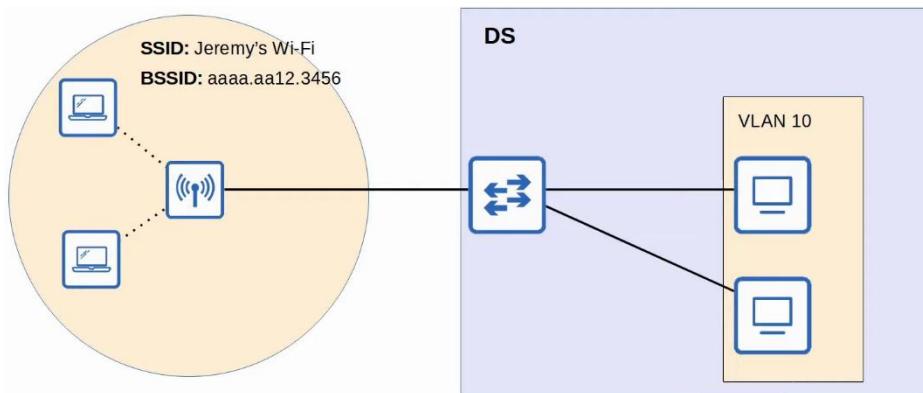
- Used in situations where it's difficult to run an Ethernet connection to every AP
- Mesh APs use 2 radios
  - 1 to provide a BSS to wireless clients
  - 1 to form a 'backhaul network' which is used to bridge traffic from AP to AP
- At least 1 AP is connected to the wired network, and it is called the RAP (Root Access Point)
- The other APs are called MAP (Mesh Access Points)
- A protocol is used to determine the best route through the mesh (similar to how dynamic routing protocols are used to determine the best path to a destination)



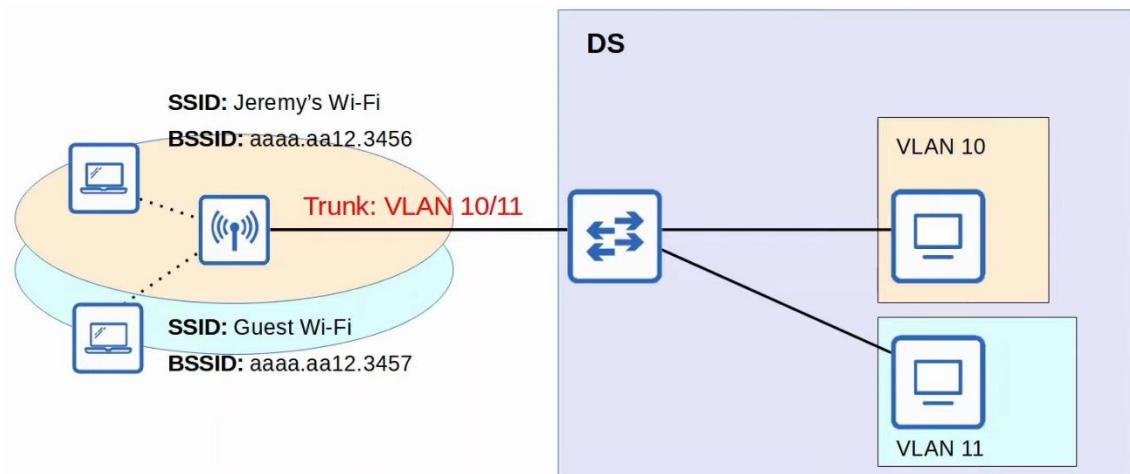
### Distribution System

- Most wireless networks aren't standalone networks

- Rather, they are a way for clients to connect to the wired network infrastructure
- In 802.11, the upstream wired network is called the DS (distribution system)
- Each wireless BSS or ESS is mapped to a VLAN in the wired network

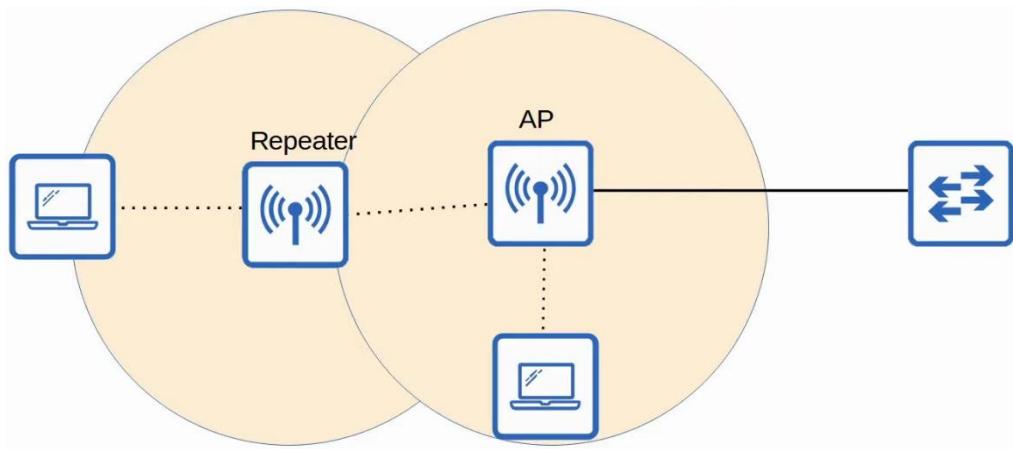


- It's possible for an AP to provide multiple wireless LANs, each with a unique SSID
- Each WLAN is mapped to a separate VLAN and connected to the wired network via a trunk
- Each WLAN uses a unique BSSID, usually by incrementing the last digit of the BSSID by 1

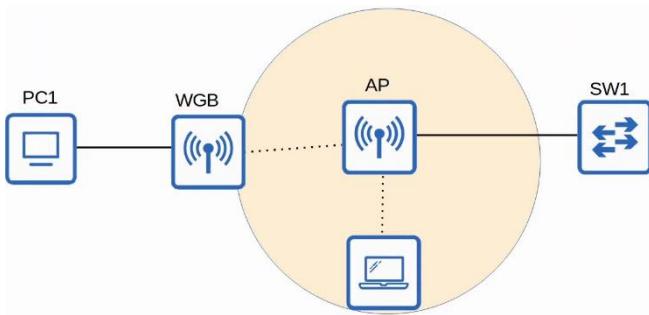


### Additional AP Operational Modes

- APs can operate in additional modes beyond the ones we have introduced so far
- An AP in repeater mode can be used to extend the range of a BSS
- The repeater will simply retransmit any signal it receives from the AP
  - A repeater with a single radio must operate on the same channel as the AP, but this can drastically reduce the overall throughput on the channel
  - A repeater with 2 radios can receive on 1 channel, and then retransmit on another channel



- A workgroup bridge (WGB) operates a wireless client of another AP, and can be used to connect wired devices to the wireless network
- In the example below, PC1 does not have wireless capabilities, and also does not have access to wired connection to SW1
- PC1 has a wired connection to the WGB, which has a wireless connection to the AP
- There are 2 kinds of WGB
  - Universal WGB (uWGB): an 802.11 standard that allows 1 device to be bridged to the wireless network
  - WGB: Cisco proprietary version of the 802.11 standard that allows multiple wired clients to be bridged to the wireless network



- Outdoor Bridge
  - Can be used to connect networks over long distances w/o a physical cable connecting them
  - The APs will use specialized antennas that focus most of the signal power in 1 direction, which allows the wireless connection to be made over longer distances than normally possible
  - The connection can be point-to-point as in the diagram below, or point-to-multipoint in which multiple sites connect to 1 central site



## Review

- Wireless LANs are defined in 802.11.
  - Operate in half duplex using CSMA/CA
- Wireless signals can be affected by absorption, reflection, refraction, diffraction, and scattering.
- Various aspects of waves can be measured, such as amplitude, frequency, and period.
- Frequency is measured in hertz (Hz).
- Wireless LANs use two frequency ranges: the 2.4 GHz band and 5 GHz band.  
-4 Wi-Fi 6 (802.11ax) can use the 6 GHz range too.
- Bands are divided into channels.
- 5 GHz band consists of non-overlapping channels.
- 2.4 GHz band channels overlap. To avoid overlapping, use channels 1, 6, and 11 (in North America).
- 802.11 standards (802.11b, 802.11a, etc) and their frequencies/theoretical max data rates.

Service sets are groups of wireless devices. Three types:

- Independent (IBSS, also called ad hoc)
- Infrastructure (BSS, ESS)

\*passing between APs in an ESS is called roaming.

- Mesh (MBSS)
- Service sets are identified by an SSID (non-unique, human-readable) and BSSID (unique, MAC address of AP).
- The area around an AP where its signal is usable is called a BSA.

- The upstream wired network is called the DS.
- When multiple WLANs are used, each is mapped to a separate VLAN on the wired network.
- APS can also operate as a repeater, workgroup bridge, or outdoor bridge.

\*Although this summarizes the topics in this video, make sure you know the details of each topic that we covered.

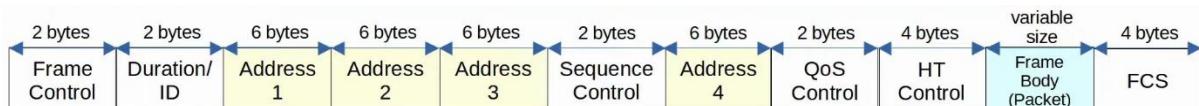
## Wireless Architecture

### Things covered

- 802.11 messages/frame format
- Autonomous APs
- Lightweight APs
- Cloud-based APs
- Wireless LAN Controller (WLC) Deployments

### 802.11 Frame Format

- Depending on the 802.11 version and messaging type, some of the fields might not be present in the frame
  - E.g. not all messages use all 4 address fields

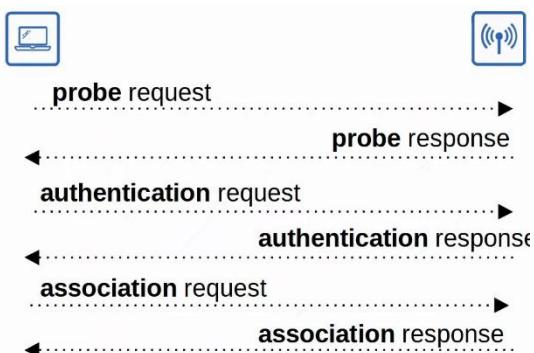


- Frame control
  - Provides information such as the message type and subtype
- Duration/ID
  - Depending on the message type, this field can indicate
    - Time (in microseconds) the channel will be dedicated for transmission of the frame
    - Identifier for the association (connection)
- Addresses
  - Up to 4 addresses can be present in an 802.11 frame
  - Which addresses are present, their order, depends on the message type
  - Destination Address (DA) - final recipient of the frame
  - Source Address (SA) - original sender of the frame

- Receiver Address (RA) - immediate recipient of the frame
- Transmitter Address (TA) - immediate sender of the frame
- Sequence Control
  - Used to reassemble fragments and eliminate duplicate frames
- HT (High Throughput) Control
  - Added in 802.11n to enable High Throughput operations
  - 802.11n - High Throughput (HT) Wi-Fi
  - 802.11ac - Very High Throughput (VHT) Wi-Fi
- FCS (Frame Check Sequence)
  - Same as in Ethernet frame, used to check for errors

## 802.11 Association Process

- Access Points bridge traffic btw wireless stations and other devices
- For a station to send traffic through the AP, it must be associated with the AP
- There are three 802.11 connection states
  - Not authenticated, not associated
  - Authenticated, not associated
  - Authenticated, associated
- The station must be fully authenticated and associated with the AP to send traffic through it
- Note: there are 2 ways for a station to scan for a BSS
  - Active scanning
    - The station sends 'probe' requests and listens for a probe response from an AP
    - Shown in example below
  - Passive scanning
    - The station listens for beacon message from an AP
    - Beacon messages are sent periodically by APs to advertise the BSS



## 802.11 Message Types

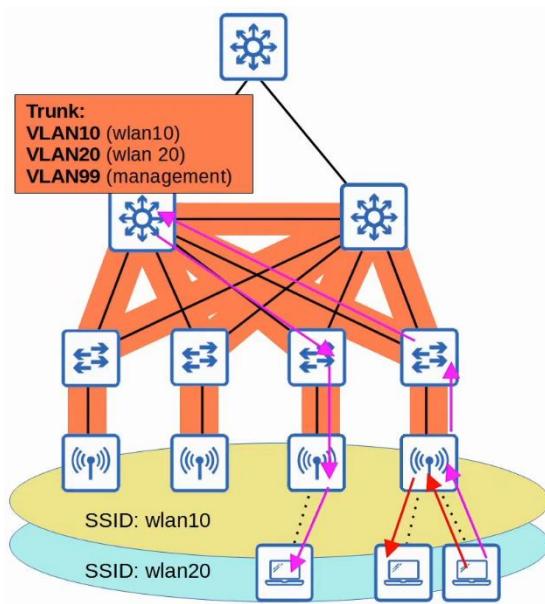
- There are 3 message types
  - Management
    - Used to manage the BSS
    - E.g. Beacon, Probe request/response, Authentication, Association
  - Control
    - Used to control access to a medium (radio frequency)
    - Assist with delivery of management and data frames
    - E.g. RTS (Request to Send), CTS (Clear to Send), ACK
  - Data
    - Used to send actual data packets

## Wireless AP Deployment

- There are 3 main wireless AP deployment methods
  - Autonomous
  - Lightweight
  - Cloud-based

## Autonomous APs

- Self-contained systems that don't rely on a WLC
- They are configured individually
  - Can be configured by console cable (CLI), telnet/SSH (CLI), or HTTP/HTTPS web connections (GUI)
  - An IP address for remote management should be configured
  - The RF parameters must be manually configured (transmit power, channel, etc)
  - Security policies are handled individually by each AP
  - QoS rules etc, are configured individually on each AP
- There is no central monitoring or management APs

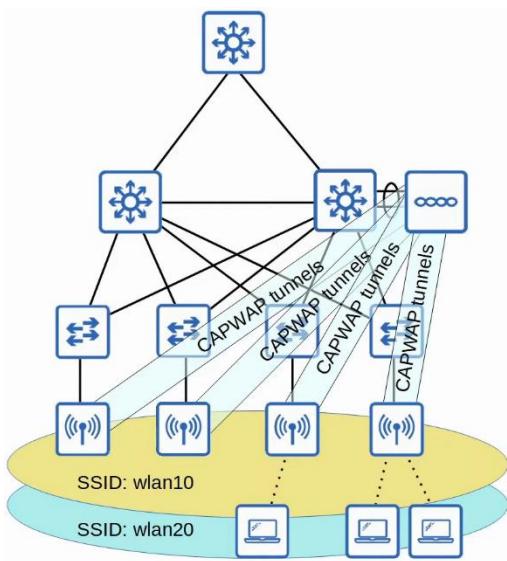


- Autonomous APs connect to the wired network with a trunk link
  - Separate management traffic with regular data traffic
- Data traffic from wireless clients has a very direct path to the wired network or to other wireless clients associated with the same AP
- Each VLAN has to stretch across the entire network, which is bad as
  - Large broadcast domains
  - Spanning tree will disable links
  - Adding/deleting VLANs is very labour-intensive
- Autonomous APs can be used in small networks, but they are not viable in medium to large networks
  - Large networks can have thousands of APs
- Autonomous APs can also function in the modes covered in the previous video
  - Repeater, Outdoor Bridge, Workgroup bridge

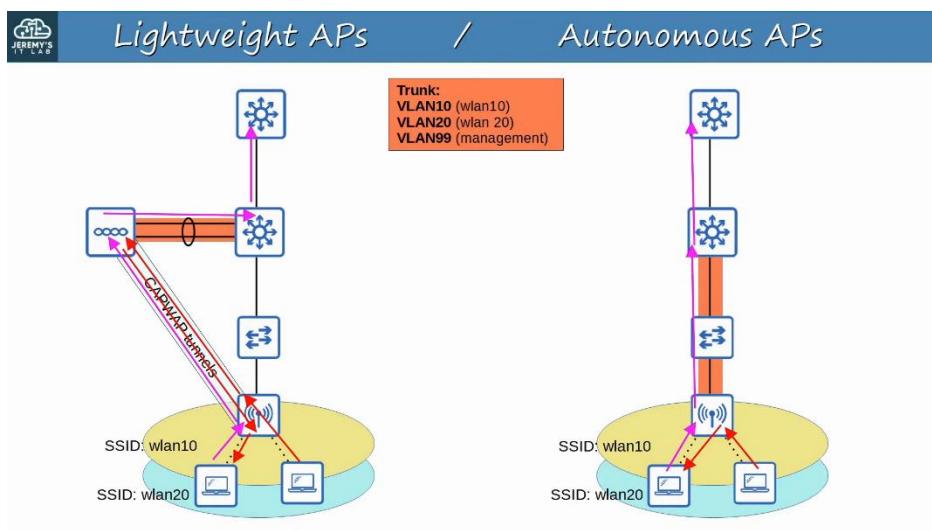
## Lightweight APs

- The functions of the AP can be split btw the AP and the WLC (Wireless LAN Controller)
- Lightweight APs handle all the real-time operations
  - E.g. transmitting/receiving RF traffic, encryption/decryption of traffic, sending out beacons/probes etc
- Other functions are carried out by a WLC
  - E.g. RF management, security/QoS management, client authentication, client association/roaming management etc
- Called 'split-MAC architecture'
- WLC also used to centrally configure the lightweight APs
- WLC can be located in the same/different subnet/VLAN as the lightweight APs it manages
- The WLC and the lightweight APs authenticate each other using digital certificates installed on each device
  - X.509 standard certificates

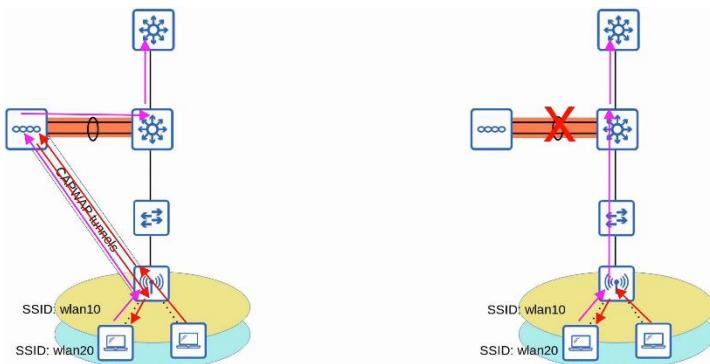
- Ensures that only authorized APs can join the network



- The WLC and lightweight APs use a protocol called CAPWAP (Control and Provisioning of Wireless Access Points) to communicate
  - Based on an older protocol called LWAPP (Lightweight Access Point Control)
- 2 tunnels are created btw each AP and the WLC
  - Control tunnel
    - UDP port 5246
    - Used to configure the APs, and control/manage the operations
    - All traffic is encrypted by default
  - Data tunnel
    - UDP port 5247
    - All traffic from wireless clients is sent through this tunnel to the WLC. It does not go directly to the wired network
    - Traffic not encrypted by default, but can configure it with encryption with DTLS (Datagram Transport Layer Security)
- Because all traffic from wireless clients is tunneled to the WLC with CAPWAP , APs connect to switch access ports, not trunk ports

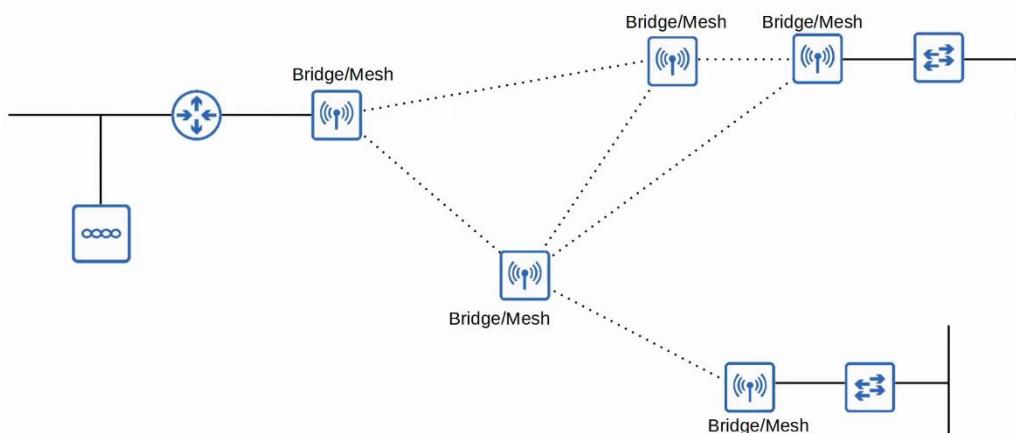


- Benefits
  - Scalability
    - With a WLC (or multiple) it's much simpler to build and support a network with thousands of APs
  - Dynamic channel assignment
    - WLC can auto select which channel each AP should use
  - Transmit power optimization
    - WLC can auto set the appropriate transmit power for each AP
  - Self-healing wireless coverage
    - When an AP stops functioning, the WLC can increase the transmit power of nearby AP's to avoid coverage holes
  - Seamless roaming
    - Clients can roam between APs with no noticeable delay
  - Client load balancing
    - If a client is in range of 2 APs, the WLC can associate the client with the least-used AP, to balance the load among APs
  - Security/QoS management
    - Central management of security and QoS policies ensures consistency across the network
- Lightweight APs can be configured to operate in different modes
  - Local
    - Default
    - AP offers a BSS (or multiple) for clients to associate with
  - FlexConnect
    - Offers 1 or more BSS for clients to associate with (like Local)
    - Allows APs to locally switch traffic btw the wired and wireless networks if the tunnels to the WLC goes down



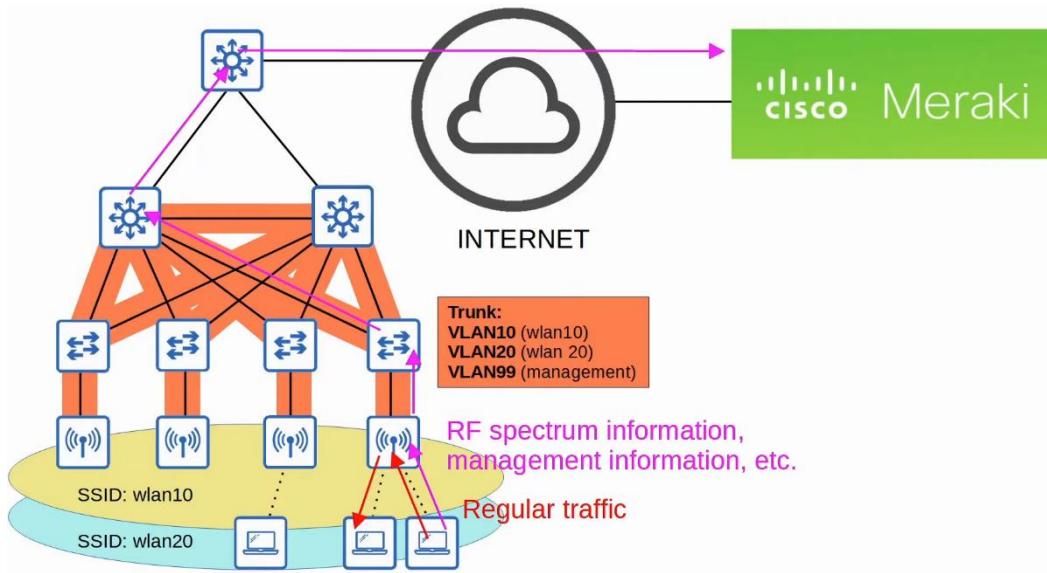
- Sniffer
  - AP does not offer BSS for clients
  - It is dedicated to capturing 802.11 frames and sending them to a device software such as Wireshark
- Monitor
  - Dedicated to receiving 802.11 frames to detect rogue devices

- If client found to be a rogue device, the AP can send de-authentication messages to disassociate the rogue device from the AP
- Rogue detector
  - AP does not use its radio
  - Listens to traffic on the wired network only
  - Receives a list of suspected rogue clients and AP MAC addresses from the WLC
  - By listening to ARP messages on the wired network and correlating it with the info it receives from the WLC, it can detect rogue devices
- SE-Connect (Spectrum Expert Connect)
  - AP does not offer BSS for clients
  - Dedicated to RF spectrum analysis on all channels
  - Can send info to software such as Cisco Spectrum Expert on a PC to collect and analyse the data
  - Can help find sources of interference
- Bridge/Mesh
  - Lightweight AP can be a dedicated bridge btw sites, e.g. over long distances
  - A mesh can be made btw the access points
- Flex plus Bridge
  - Adds FlexConnect functionality to the Bridge/Mesh mode
  - Allows wireless access points to locally forward traffic even if connectivity to the WLC is lost



## Cloud-based APs

- Cloud-based AP architecture is in btw autonomous AP and split-MAC architecture
  - Autonomous APs that are centrally managed in the cloud
- Cisco Meraki is a popular cloud-based Wi-Fi solution
- The Meraki dashboard can be used to configure APs , monitor the network, generate performance reports, etc
  - Meraki also tells each AP which channel to use, what transmit power, etc
- However, data traffic is not sent to the cloud
  - It is sent directly to the wired network like when using autonomous APs
  - Only management/control traffic is sent to the cloud

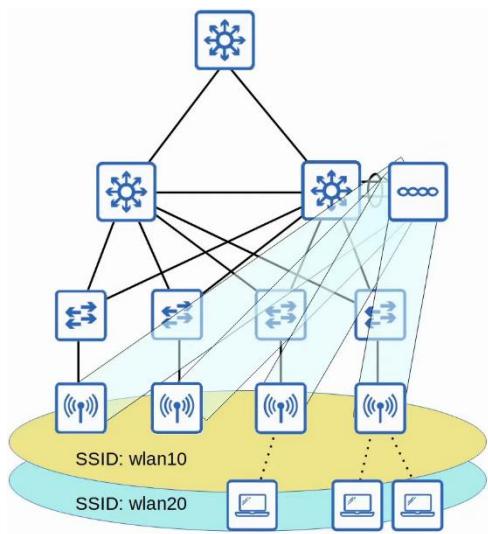


## WLC Deployment Models

- In a split-MAC architecture, there are 4 main WLC deployment models
  - Unified
    - The WLC is a hardware appliance in a central location of the network
  - Cloud-based
    - The WLC is a VM running on a server, usually in a private cloud in a data center
    - This is not the same as the cloud-based AP architecture discussed previously
  - Embedded
    - The WLC is integrated within a switch
  - Mobility Express
    - The WLC is integrated within an AP

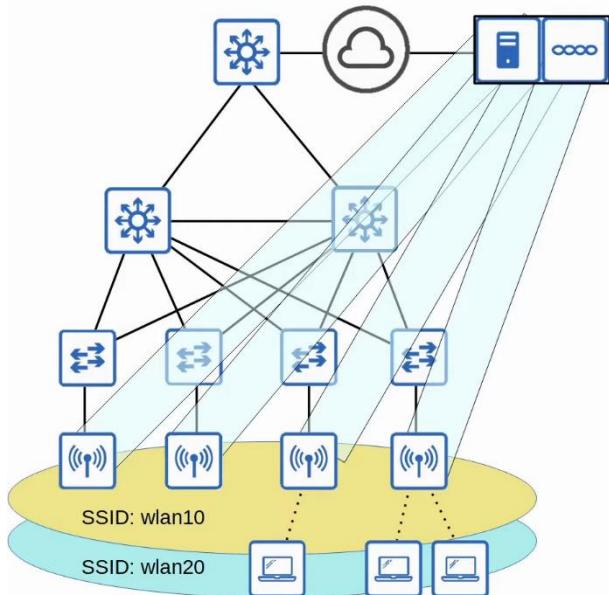
## Unified WLC

- Hardware appliance deployed in a central location of the network
- Can support up to 6000 APs
- If more APs needed, additional WLCs can be added to the network



### Cloud-based WLC

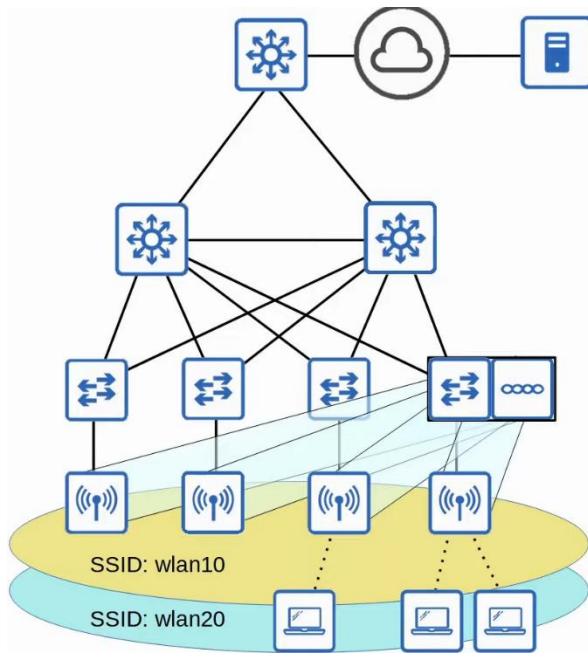
- WLC is a VM running on a server, typically in a private cloud or data center
- Can support up to 3000 APs
- If more APs needed, more WLCs can be added



### Embedded WLC

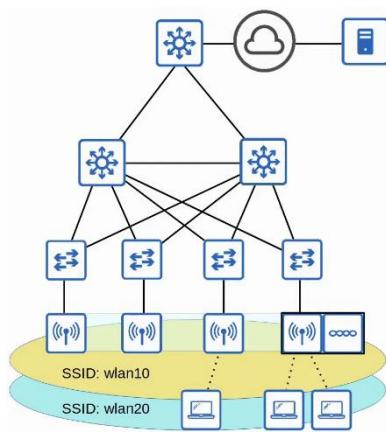
- Embedded within a switch

- Can support up to 200 APs
- If more APs needed, can add more switches with embedded WLC



### Cisco Mobility Express WLC

- Embedded within an AP
- Support up to 100 APs
- If more needed, can add more APs with embedded Mobility Express WLC



# Wireless Security

## Things Covered

- Intro
- Authentication Methods
- Encryption/Integrity methods
- Wi-Fi Protected Access (WPA)

## Intro

- More important in wireless network
  - Wireless signals not contained within wire, any device within signal range can receive traffic
- Encryption
  - Wired networks encrypted only when sent over untrusted networks e.g. Internet
  - Wireless networks, important to encrypt traffic btw wireless clients and AP
- 3 main concepts
  - Authentication
  - Encryption
  - Integrity

## Authentication

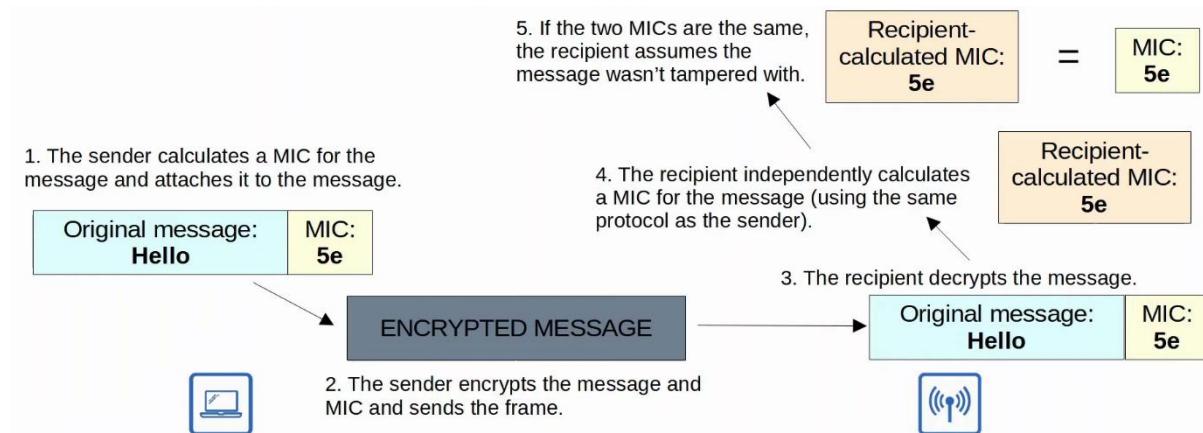
- All clients must be authenticated before they can associate with AP
- In corporate settings, only trusted devices should be given access to network
  - Separate SSID which doesn't have access to corporate network can be used for guests
- Clients should also authenticate AP to avoid associating with malicious AP
- Multiple ways to authenticate
  - Password
  - Username / Password
  - Certificates

## Encryption

- Any wireless traffic should be encrypted
- Many protocols
  - Both client and AP need to have the same protocol
- All devices in WLAN will use same protocol
  - However, each client will use a unique encryption/decryption key
- 'Group key' used by AP to send to all clients
  - All clients associated with AP will also have the key

## Integrity

- Ensures that traffic is not modified in transit
- MIC (Message Integrity Check) is added to check the integrity



## Authentication Methods

- Open Authentication
- WEP (Wired Equivalent Privacy)
- EAP (Extensible Authentication Protocol)
- LEAP (Lightweight EAP)
- EAP-FAST (EAP Flexible Authentication via Secure Tunnelling)
- PEAP (Protected EAP)
- EAP-TLS (EAP Transport Layer Security)

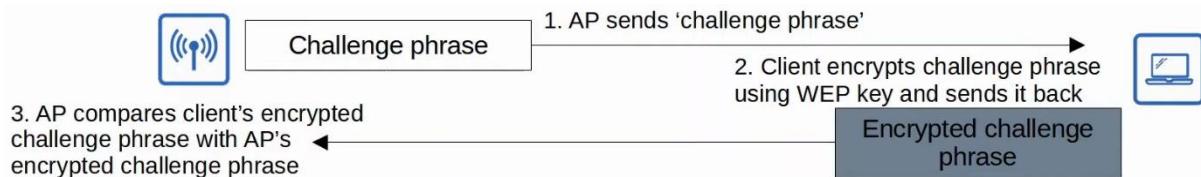
## Open Authentication

- Client sends authentication request, AP accepts it, no questions asked
- Not secure
- Possible after authentication, user needs to authenticate via another method before granted access to network
- E.g. Airport Wi-Fi

## WEP (Wired Equivalent Privacy)

- Provides authentication and encryption
- Encryption
  - RC4 algorithm
  - 'Shared-key' protocol, AP and client have same key

- Key: 40/104 bits + 24 bit IV (Initialisation Vector)
  - Total: 64/128 bits
- Not secure, easily cracked
- Authentication

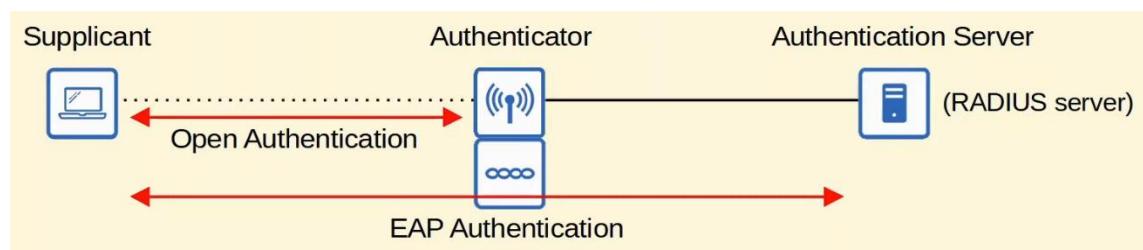


### EAP (Extensible Authentication Protocol)

- It is an authentication framework
- Defines a standard set of authentication functions that are used by various EAP methods
- EAP is integrated with 802.1X, which provides port-based network access control

### 802.1X

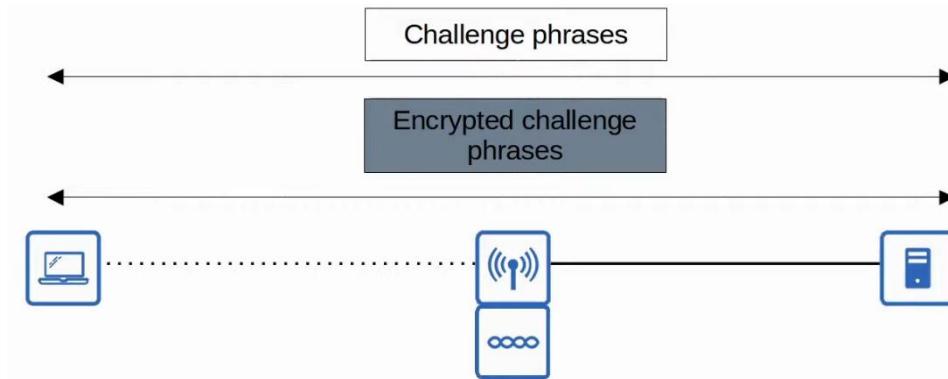
- Used to limit network access for clients connected to a LAN or WLAN until they authenticate
- 3 main entities
  - Supplicant
    - Device that wants to connect to the network
  - Authenticator
    - Device that provides the access to the network
  - Authentication Server
    - Device that receives client credentials and permit/denies access
- When connected through open authentication, only EAP authentication allowed, all other traffic not allowed



### LEAP (Lightweight EAP)

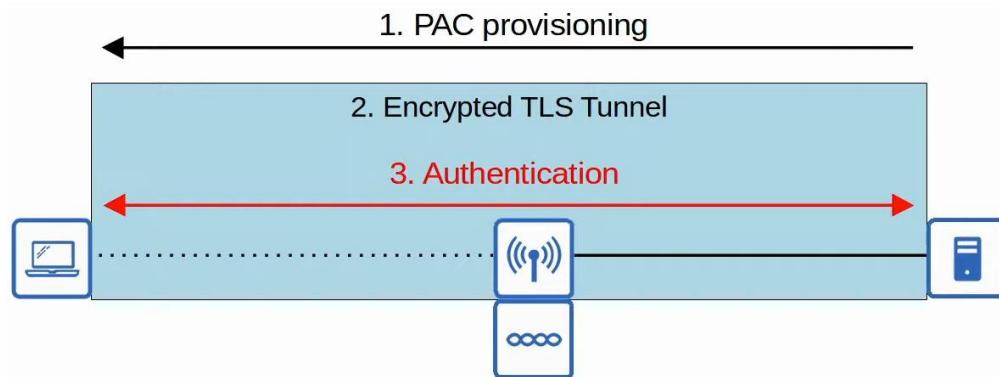
- Developed by Cisco as upgrade to WEP
- Client must provide username and password to authenticate

- In addition, mutual authentication is provided by both the client and server sending a challenge phrase to each other
- Dynamic WEP keys are used, meaning the WEP keys are changed frequently
- Considered vulnerable and should not be used



### EAP-FAST (EAP Flexible Authentication via Secure Tunnelling)

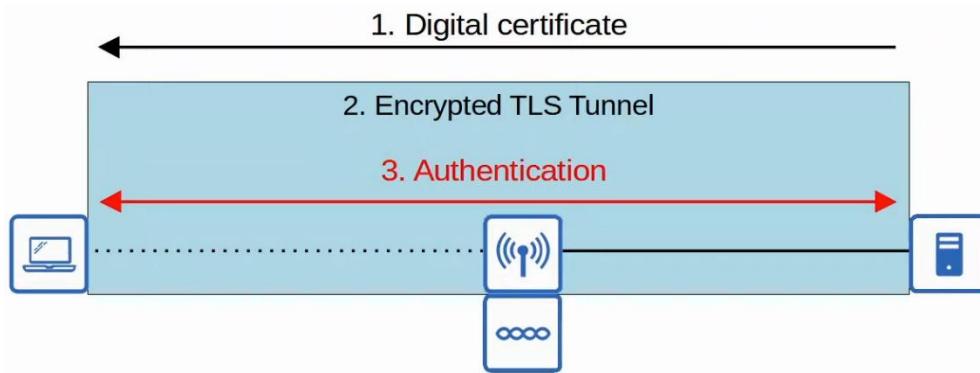
- Developed by Cisco
  - 3 phases
1. A PAC (Protected Access Credential) is generated and passed from server to client
  2. A secure TLS Tunnel is established btw the client and authentication server
  3. Inside of the secure (encrypted) TLS tunnel, the client and server communicate further to authenticate/authorize the client



### PEAP (Protected EAP)

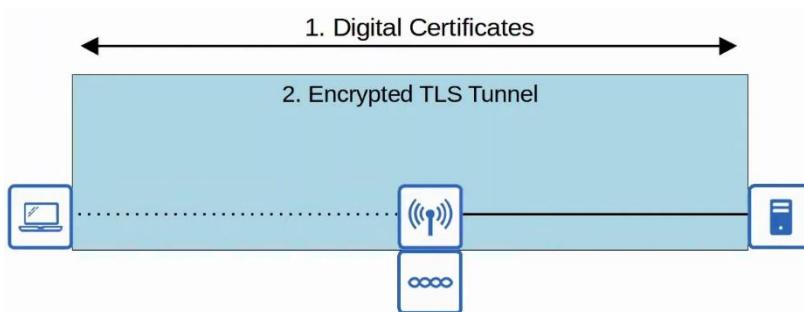
- Like EAP-FAST, involves establishing a secure TLS tunnel btw client and server
- Use digital certificate instead of PAC
  - Used by client to authenticate server
  - Used to establish TLS tunnel

- Because only the server provides a certificate for authentication, the client must still be authenticated within the secure tunnel
  - E.g. using MS-CHAP (Microsoft Challenge-Handshake Authentication Protocol)



### EAP-TLS (EAP Transport Layer Security)

- AS (Authentication Server) and all clients need to have digital certificates
- Most secure, but difficult to implement as all clients need to have cert
- Since client and server authenticate each other with digital cert, no need to authenticate client within the TLS tunnel
- TLS tunnel still used to exchange encryption key information



### Encryption/Integrity Methods

- TKIP (Temporal Key Integrity Protocol)
- CCMP (Counter/CBC-MAC Protocol)
- GCMP (Galois/Counter Mode Protocol)

### Temporal Key Integrity Protocol (TKIP)

- Temporary solution for WEP since most devices were built for WEP

- Added security features
  - MIC added
  - Key mixing algorithm used to create unique WEP keys for every frame
  - Initialization vector doubled to 48 bits, making brute-force attacks more difficult
  - MIC includes sender MAC address
  - Timestamp added to MIC to prevent replay attacks
    - Replay attacks involve sending a frame that has already been transmitted
  - TKIP sequence number used to keep track of frames sent from each source MAC address
    - Also protects against replay attacks
- Note:
  - Used in WPA version 1
  - Just know that it is an upgrade to WEP

### **Counter/CBC-MAC Protocol (CCMP)**

- Developed after TKIP, more secure
- WPA2
- Must be supported by hardware
- 2 different algorithm for encryption and MIC
  - AES (Advanced Encryption Standard)
    - 'Counter' mode
    - Most secure encryption in the world
    - Multiple modes, CCMP uses 'counter'
  - CBC-MAC (Cipher Block Chaining Messages Authentication Code)
    - Used as MIC

### **Galois/Counter Mode Protocol (GCMP)**

- More secure and efficient than CCMP
- Higher data throughput than CCMP
- WPA3
- 2 algorithms
  - AES counter mode
  - GMAC (Galois Message Authentication Code)
    - MIC

### **Wi-Fi Protected Access**

- Wi-Fi Alliance has developed 3 WPA certifications for wireless devices
  - WPA
  - WPA2
  - WPA3
- To be WPA-certified, equipment must be tested and authorized
- All WPA support 2 authentication modes

- Personal mode
  - Pre-shared key (PSK) used for authentication
  - E.g. connect to home wifi, enter the password and are authenticated
  - Common for small networks
  - Note: PSK not sent over air. A 4-way handshake is used for authentication, and the PSK is used to generate the encryption keys
- Enterprise mode
  - 802.1X is used with an authentication server (RADIUS server)
  - \*No specific EAP method specified, all are supported

## **WPA**

- Developed after WEP proven to be vulnerable
- Consist of the following protocols
  - TKIP (based on WEP) provides encryption/MIC
  - 802.1X authentication (Enterprise mode) or PSK (Personal mode)

## **WPA2**

- Released 2004
- Protocols
  - CCMP provides encryption/MIC
  - 802.1X authentication (Enterprise mode) or PSK (Personal mode)

## **WPA3**

- Released 2018
- Protocols
  - GCMP provides encryption/MIC
  - 802.1X authentication (Enterprise mode) or PSK (Personal mode)
  - Have additional security features
    - PMF (Protected Management Frames)
      - Protect 802.11 frames from eavesdropping/forging
    - SAE (Simultaneous Authentication of Equals)
      - Protects the 4-way handshake when using personal mode authentication
    - Forward Secrecy
      - Prevents data from being decrypted after it has been transmitted over the air
      - An attacker can't capture wireless frames and then try to decrypt them later

# Network Automation

## Network Automation

Things covered

- Why network automation
- Benefits
- Logical 'planes' of network functions
- Software-defined networking (SDN)
- APIs
- Data Serialization

## Network Automation

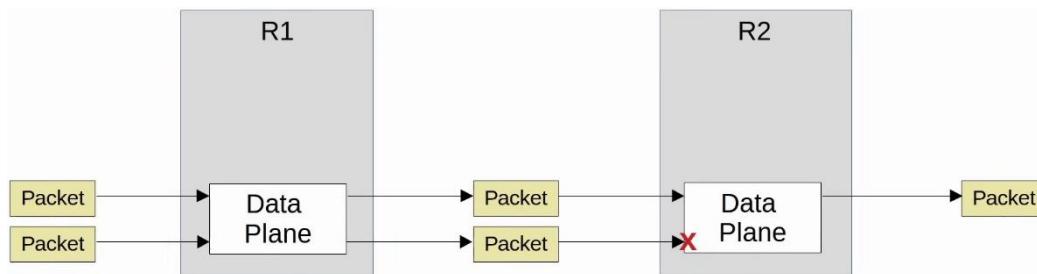
- In traditional model, engineers manage devices one at a time by connecting to their CLI
- Disadvantages
  - Typos and small mistakes are common
  - It is time-consuming and very inefficient in large-scale networks
  - Difficult to ensure all devices adhere to the organization's standard configurations
- Network automation provides many benefits
  - Human error reduced
  - Networks become much more scalable. New deployments, network-wide changes, and troubleshooting can be implemented in a fraction of the time
  - Network wide policy compliance can be assured
  - The improved efficiency of network operations reduces the opex (operation expenses) of the network. Each task requires fewer man-hours
- There are various tools/methods that can be used to automate tasks in the network
  - SDN (Software-Defined Networking)
  - Ansible
  - Puppet
  - Python scripts
  - Etc

## Logical Planes

- The various functions of network devices can be logically divided into planes
  - Data plane
  - Control plane
  - Management plane

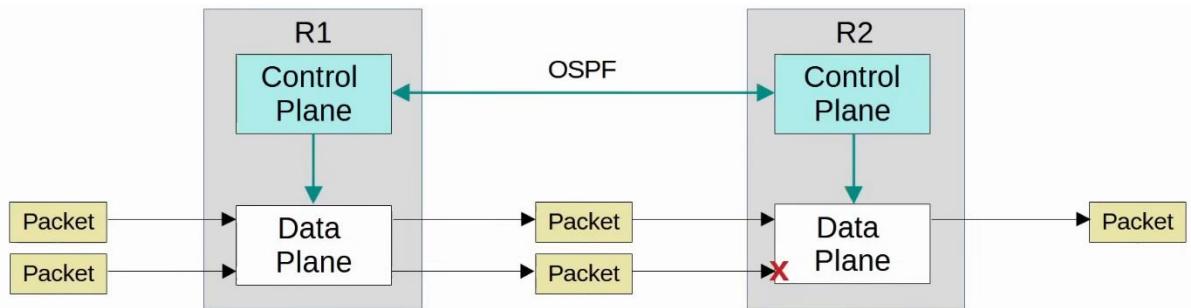
## Data Plane

- All tasks involved in forwarding user data/traffic from 1 interface to another are part of the data plane
- Also called 'forwarding plane'
- Example
  - A router receives a message, looks for the most specific matching route in its routing table, and forwards it out of the appropriate interface to the next hop
    - It also de-encapsulates the original Layer 2 header, and re-encapsulates with a new header destined for the next hop's MAC address
  - A switch receives a message, looks at the destination MAC address, and forwards it out of the appropriate interface (or floods it)
    - This includes functions like adding or removing 802.1Q VLAN tags
  - NAT (changing the src/dst address before forwarding) is part of the data plane
  - Deciding to forward or discard messages due to ACLs, port security, etc is part of the data plane



## Control Plane

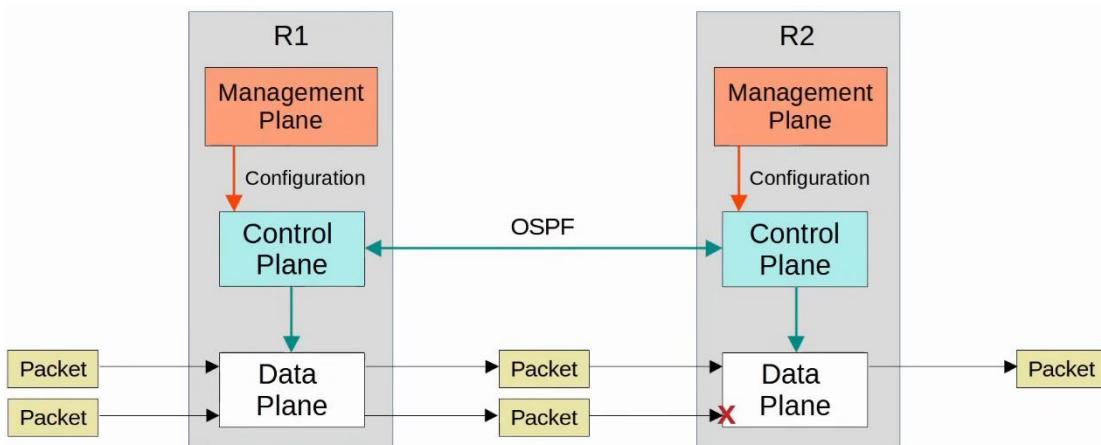
- How does a device's data plane make its forwarding decisions?
  - Routing table, MAC address table, ARP table, STP, etc
- Functions that build these tables (and other functions that influence the data plane) are part of the control plane
- The control plane controls what the data plane does, e.g. by building the router's routing table
- The control plane performs overhead work
  - E.g. OSPF itself does not forward user packet, but it informs the data plane how packets should be forwarded
  - E.g. STP - which interface should/shouldn't forward frames
  - E.g. ARP - build ARP table that is used in the process of forwarding data



- In traditional networking, the data plane and control plane are both distributed
  - Each device has its own data and control plane
  - The planes are 'distributed' throughout the network

### Management Plane

- Performs overhead work
- Doesn't directly affect the forwarding of messages in the data plane
- Consists of protocols that are used to manage devices
  - SSH/Telnet - used to connect to the CLI of the device
  - Syslog - used to keep logs of events that occur on the device
  - SNMP - used to monitor the operations of the device
  - NTP - used to maintain accurate time on the device



- The data plane is the reason why we buy routers and switches (and network infrastructure in general), to forward messages
  - However, the Control plane and Management plane are both necessary to enable the data plane to do its job

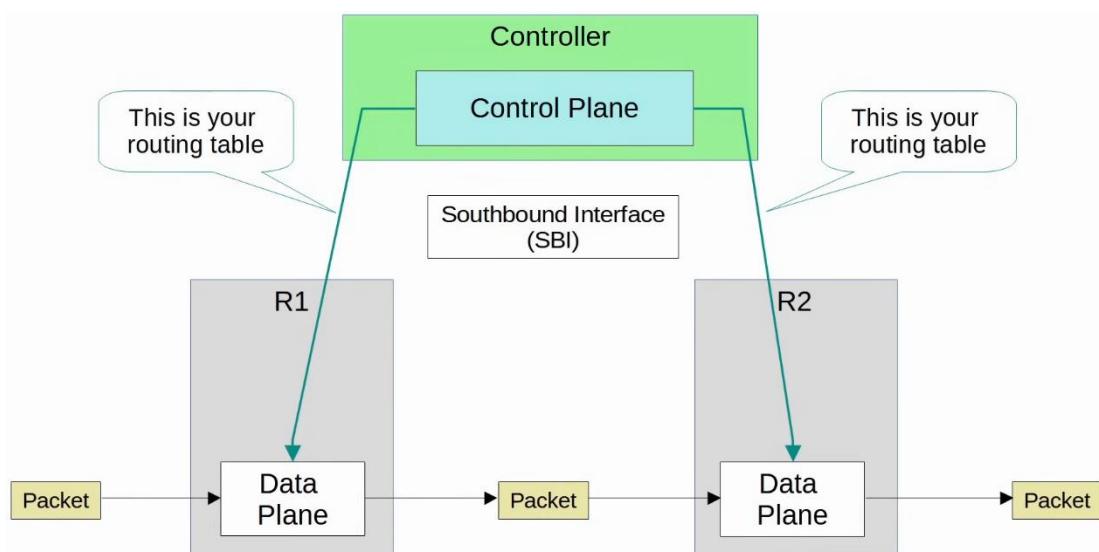
### Hardware Management

- Operations of the Management and Control plane are usually managed by the CPU
- However, not desirable for Data plane operations since CPU processing is slow

- ASIC (Application-Specific Integrated Circuit) used
- Using switch for example
  - When frame is received, ASIC responsible for switching logic
  - MAC address table stored in TCAM (Ternary Content-Addressable Memory)
  - MAC address table also called CAM table
  - ASIC feeds the destination MAC address of frame into TCAM, which returns the matching MAC address table entry
  - The frame is then forwarded out of the appropriate interface
- Modern routers also use a similar hardware data plane
  - ASIC for forwarding logic
  - Tables stored in TCAM
- Simple summary
  - When a device receives control/management traffic (destined for itself), it will be processed by CPU
  - When a device receives data traffic which should pass through the device, it is processed by the ASIC for max speed

## Software-Defined Networking (SDN)

- An approach to networking that centralizes the control plane into an application called a 'controller'
- SDN also called Software-Defined Architecture (SDA) or Controller-Based Networking
- Traditional control planes use a distributed architecture
  - E.g. each router in the network runs OSPF and the routers share routing information and then calculate their preferred routes to each destination
- An SDN controller centralizes control plane functions (e.g. calculating routes)
  - That is just an example, how much of the control plane is centralized varies greatly
- The controller can interact programmatically with the network devices using APIs (Application Programming Interface)

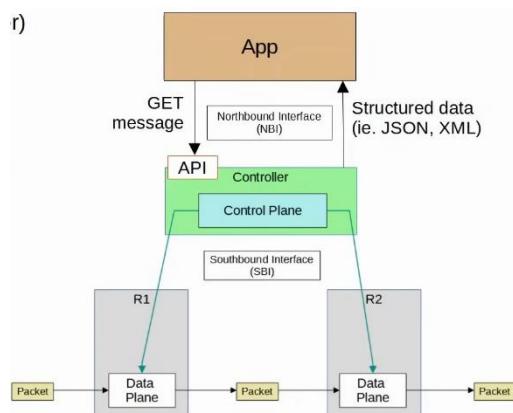


## Southbound Interface (SBI)

- Used for communication btw controller and network devices it controls
- Typically consists of communication protocol and API
- APIs facilitate data exchanges btw programs
  - Data is exchanged btw controller and network devices
  - An API on the network device allows the controller to access information on the device, control their data plane tables, etc
- Some examples of SBIs
  - OpenFlow
  - Cisco OpFlex
  - Cisco onePK (Open Network Environment Platform Kit)
  - NETCONF

### **Northbound Interface (NBI)**

- Allows us (user) to interact with the controller, access data it gathers about the network, program it, and make changes in the network via SBI
- A REST API is used on the controller as an interface for apps to interact with it
  - REST - Representational State Transfer
- Data is sent in a structured (serialized) format such as JSON/XML
  - Makes it easier for programs to use the data



### **Automation in Traditional Networks vs SDN**

- Traditional Networks
  - Scripts can be written (e.g. using Python) to push commands to many devices at once
  - Python with good use of 'Regular Expressions' can parse through "show" commands to gather information about the network devices
- However, the robust and centralized data collected by SDN controllers greatly facilitates these functions
  - Controller collects information about all devices in the network

- Northbound APIs allow apps to access information in a format that is easy for programs to understand (e.g. JSON, XML)
  - The centralized data facilities network-wide analytics
- SDN tools provide the benefits of automation w/o the requirement of 3rd party scripts and apps
  - Don't need to have expertise in automation to make use of SDN tools
  - However, APIs allow 3rd party applications to interact with the controller, which can be very powerful
- Note: Although SDN and automation aren't the same thing, the SDN architecture greatly facilitates the automation of various tasks in the network via the SDN controller and APIs

## AI & Machine Learning

### Things covered

- What are AI and ML
- Types of ML
- Predictive/Gen AI
- AI in Catalyst Center

### What is AI

- AI uses computers to simulate intelligence, allowing them to exhibit behaviour typically associated with humans
- Examples
  - Virtual assistants: Siri, Alexa
  - Recommendation systems: Netflix, YouTube
  - Self-driving cars: Tesla
  - Etc

### Machine Learning (ML)

- Subset of AI
- Focus on enabling computers to learn from data and improve w/o the need for explicit programming
  - Instead of hard coded instructions, ML algorithms identify patterns in data and make predictions or decisions based on those pattern

### Types of ML

- Supervised Learning
  - Model trained on labelled data, correct answers provided
  - Make predictions or classification on new data
- Unsupervised Learning
  - Model given unlabelled data
  - Tasked with finding patterns, relationships, or groupings within the data

- Reinforcement Learning
  - Model learns by interacting with an environment
  - Receives rewards/penalties based on its actions to maximise its performance over time
- Deep Learning
  - Specialized subset of ML that uses multi-layered neural network to handle large datasets and perform complex tasks like image recognition and natural language processing

## **Supervised Learning**

- Train on labelled dataset
- Advantage
  - Highly accurate when labelled data available
  - Easy to understand and deploy
- Disadvantage
  - Require large datasets
  - Output is restricted to the labelled data
    - E.g. can only label cat and dog, cannot label bird

## **Unsupervised Learning**

- Train on unlabelled dataset
- Groups similar data based on similar features
- Advantage
  - No need for labelled data
  - Reveals hidden patterns
- Disadvantage
  - Interpretation and labelling of results required (human input)
  - Less accurate

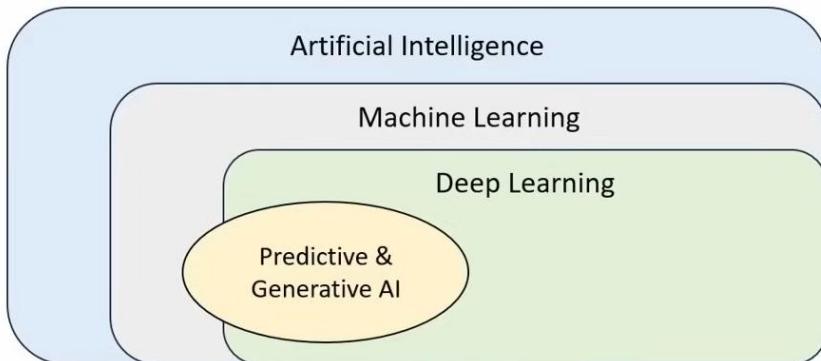
## **Reinforcement Learning**

- Train a model by rewarding/penalizing its action in a given environment to maximise performance over time
- How it works
  - Model (Agent) interacts w environment
  - Takes action and receives feedback
  - Over time, learns which actions give the best results
- E.g. chess engine
- Advantage
  - Capable of learning complex behaviours
  - Adapts to dynamic environments

- Disadvantage
  - Resource intensive
  - Risk of suboptimal results if not designed properly
    - E.g. choose short-term results over long-term

## Deep Learning

- Use artificial neural network
- Data pass through multiple layers of nodes (neurons)
- Can be trained using supervised, unsupervised and reinforcement methods
- Advantage
  - Excels at handling large, unstructured datasets (e.g. image/audio/text)
  - Achieves state-of-the-art performance in tasks like image recognition, natural language processing, etc
- Disadvantage
  - Resource intensive
  - "Black box" difficult to interpret how it get its results



## Predictive & Generative AI

- Predictive AI
  - Use machine learning to analyse historical data and predict trends
    - E.g. weather forecasting, security anomaly detection
- Generative AI
  - Use machine learning to learn patterns from existing data and create new content

## Predictive AI

- Applications
  - Healthcare
  - Network security
  - Traffic management
  - Business forecasting
  - Weather forecasting
- Advantage
  - Improves decision-making by providing actionable insights
  - Detects potential problems before they occur
- Disadvantage
  - Requires high-quality, relevant historical data
  - Accuracy depends on how well the patterns in data generalize to new scenarios

## **Generative AI**

- Applications
  - Text generation
  - Image generation
  - Video generation
- Advantage
  - Great for creative tasks where human input is limited/time-consuming
  - Enables automation of content creation across various fields
- Disadvantage
  - Risk of misuse (e.g. deepfakes)
  - Generated content is only as good as the quality of training material
  - Hallucinations (give wrong answers)

## **Predictive & Generative AI in networking**

- Predictive AI
  - Traffic Forecasting
  - Security threat detection
  - Predictive maintenance
- Generative AI
  - Network documentation
  - Config generation
  - Network design
  - Troubleshooting
  - Script generation

## **AI in Cisco Catalyst Center**

- Cisco Catalyst Center (formerly DNA Center) features a variety of AI-enabled features to identify issues before they impact users, reduce the time required to solve issues, and increase the performance and security of the network
- Features include
  - AI Network Analytics
    - Use AI to establish baseline behaviour
    - Provides insights and recommendations for optimizing network performance
    - Continuously monitors the network to detect and predict anomalies
  - Machine Reasoning Engine
    - Use AI to find root cause of issue
    - Suggest methods to solve the issue / auto resolve issue
  - AI Endpoint Analytics
    - Identify and classify devices on network
    - Detects unauthorized devices / unusual behaviour
    - Simplifies onboarding by automating profiling and segmentation
  - AI-enhanced Radio Resource Management
    - Manage Access Points

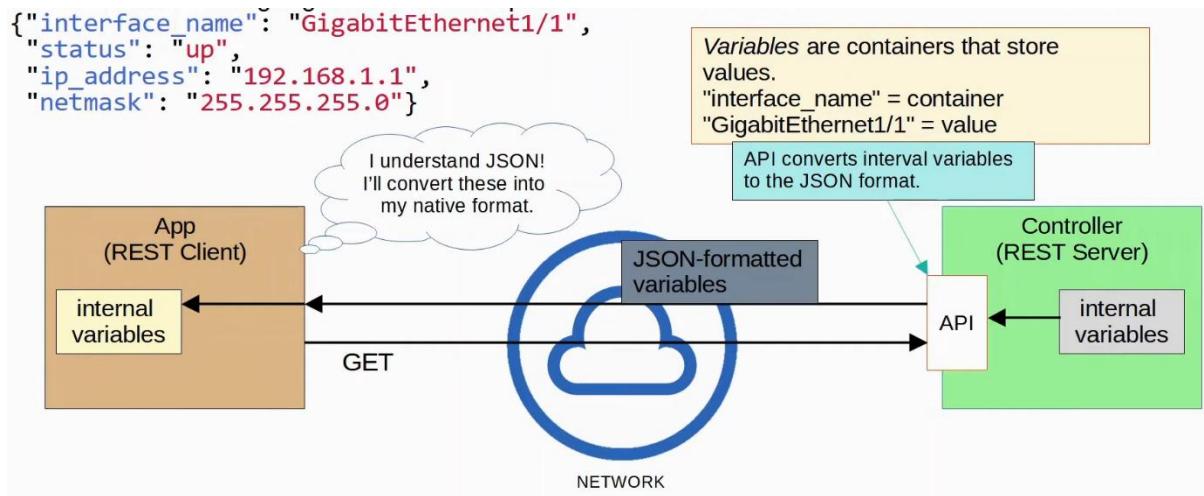
## JSON, XML, YAML

Things covered

- Data serialization
- JSON (JavaScript Object Notation)
- XML (Extensible Markup Language)
- YAML (YAML Ain't Markup Language)

### Data Serialization

- Process of converting data into a standardized format that can be stored (in a file) or transmitted (over network) and reconstructed later (i.e. by a different application)
  - Allows the data to be communicated btw applications in a way both applications understand
- Data serialization languages allow us to represent variables with text



## JSON

- An open standard file format and data interchange format that uses human-readable text to store and transmit data objects
- RFC 8259
- Whitespace insignificant
- 4 primitive data types
  - String
  - Number
  - Boolean: true/false, no quotes ("")
  - Null: null, no quotes
- 2 structured data types
  - Object/Dictionary: unordered list of key-value pair
  - Array: values don't have to be same types

R1#show ip interface brief						
Interface	IP-Address	OK?	Method	Status	Protocol	
GigabitEthernet0/0	192.168.1.1	YES	manual	up	up	
GigabitEthernet0/1	unassigned	YES	unset	administratively down	down	

```
{
  "ip_interfaces": [
    {
      "Interface": "GigabitEthernet0/0",
      "IP-Address": "192.168.1.1",
      "OK?": "YES",
      "Method": "manual",
      "Status": "up",
      "Protocol": "up"
    },
    {
      "Interface": "GigabitEthernet0/1",
      "IP-Address": "unassigned",
      "OK?": "YES",
      "Method": "unset",
      "Status": "administratively down",
      "Protocol": "down"
    }
  ]
}
```

## XML

- Developed as markup language, but now commonly used as data serialization language
  - Markup language (e.g. HTML) used to format text
- Whitespace insignificant
- Often used by REST APIs
- <key>value</key>

```
R1#show ip interface brief
Interface                  IP-Address      OK? Method Status          Protocol
GigabitEthernet0/0          192.168.1.1   YES manual up             up
GigabitEthernet0/1          unassigned    YES unset  administratively down down

R1#show ip interface brief | format
<?xml version="1.0" encoding="UTF-8"?>
<ShowIpInterfaceBrief xmlns="ODM://built-in//show_ip_interface_brief">
  <SpecVersion>built-in</SpecVersion>
  <IPInterfaces>
    <entry>
      <Interface>GigabitEthernet0/0</Interface>
      <IP-Address>192.168.1.1</IP-Address>
      <OK>YES</OK>
      <Method>manual</Method>
      <Status>up</Status>
      <Protocol>up</Protocol>
    </entry>
    <entry>
      <Interface>GigabitEthernet0/1</Interface>
      <OK>YES</OK>
      <Method>unset</Method>
      <Status>administratively down</Status>
      <Protocol>down</Protocol>
    </entry>
  </IPInterfaces>
</ShowIpInterfaceBrief>
```

## YAML

- Yet Another Markup Language, but to distinguish it as a data serialization language
  - YAML Ain't Markup Language
- Used by Ansible
- Human-readable
- Whitespace significant
- File starts with "---
- "-" used to indicate a list
- Keys and values represented as
  - Key:value

```
---  
ip_interfaces:  
- Interface: GigabitEthernet0/0  
  IP-Address: 192.168.1.1  
  OK?: 'YES'  
  Method: manual  
  Status: up  
  Protocol: up  
- Interface: GigabitEthernet0/1  
  IP-Address: unassigned  
  OK?: 'YES'  
  Method: unset  
  Status: administratively down  
  Protocol: down
```

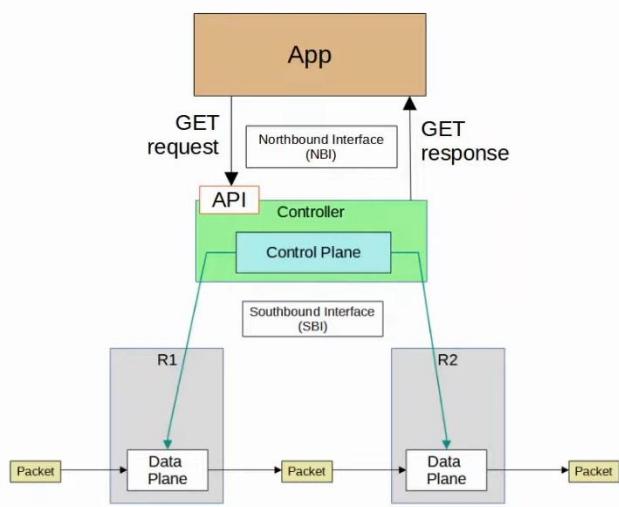
## REST APIs

### Things covered

- API review
- CRUD operations and HTTP verbs
- REST APIs
- REST API Calls using Cisco DevNet

### API (Application Programming Interface)

- Software interface that allows 2 applications to communicate
- Essential not just for automation, but for all application
- In SDN architecture, APIs used to communicate btw apps and the SDN controller (via NBI), and btw SDN controller and network devices (via SBI)
- NBI usually use REST APIs
- NETCONF and RESTCONF are popular southbound APIs



## CRUD

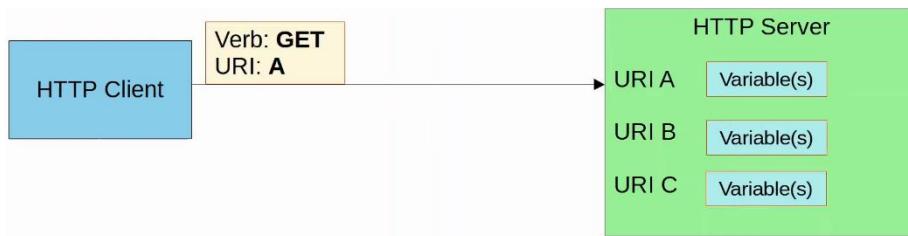
- Create, Read, Update, Delete
- Operations performed using REST APIs
- HTTP uses verbs (aka methods) that map to these CRUD operations
- REST APIs typically use HTTP

## HTTP Verbs

Purpose	CRUD Operation	HTTP Verb
Create new variable	Create	POST
Retrieve value of variable	Read	GET
Change the value of variable	Update	PUT, PATCH
Delete variable	Delete	DELETE

## HTTP Request

- When a HTTP client sends a request to an HTTP server, the HTTP header includes information like this:
  - A HTTP Verb (e.g. GET)
  - An URI (Uniform Resource Identifier), indicating the resource it is trying to access

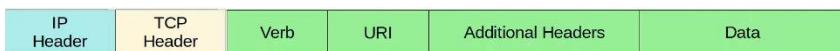


- Example URI

`https://sandboxdnac.cisco.com/dna/intent/api/v1/network-device`

scheme	authority	path
--------	-----------	------

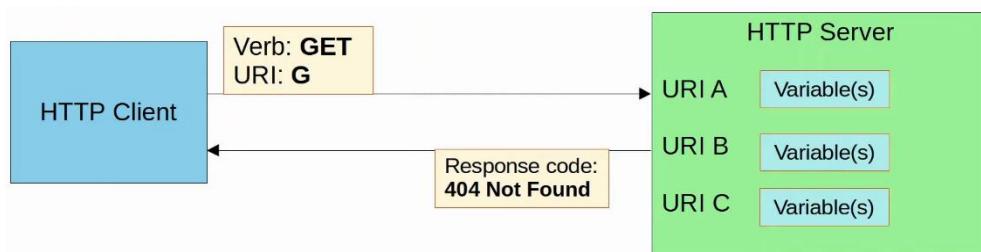
- The HTTP request can include additional headers which pass additional information to the server



- An example would be an Accept header, which informs the server about the type(s) of data that can be sent back to the client
  - E.g. `Accept: application/json` or `Accept: application/xml`
- When a REST client makes an API call (request) to a REST server, it will send a HTTP request like the one above
- Note:
  - \*REST APIs don't have to use HTTP for communication, although HTTP is the most common choice
  - REST is a framework

## HTTP Response

- Will include a status code indicating if the request succeeded or failed, as well as other details
- The first digit indicates the class of the response
  - 1xx informational : the request was received, continuing process
  - 2xx successful : the request was successfully received, understood, and accepted
  - 3xx redirection : further action needs to be taken in order to complete the request
  - 4xx client error: the request contains bad syntax or cannot be fulfilled
  - 5xx server error: the server failed to fulfil an apparently valid request



## Example

Here are some examples of each HTTP Response class:

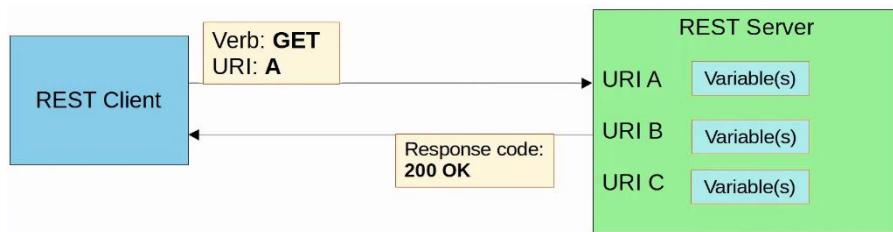
- **1xx Informational**
  - **102 Processing** indicates that the server has received the request and is processing it, but the response is not yet available.
- **2xx Successful**
  - **200 OK** indicates that the request succeeded.
  - **201 Created** indicates that the request succeeded and a new resource was created (ie. in response to POST)
- **3xx Redirection**
  - **301 Moved Permanently** indicates that the requested resource has been moved, and the server indicates its new location.
- **4xx Client Error**
  - **403 Unauthorized** means the client must authenticate to get a response.
  - **404 Not Found** means the requested resource was not found.
- **5xx Server Error**
  - **500 Internal Server Error** means the server encountered something unexpected that it doesn't know how to handle.

## REST (Representational State Transfer)

- Aka REST-based APIs or RESTful APIs
- REST isn't a specific API, instead, it describes a set of rules about how the API should work
- The 6 constraints of RESTful architecture
  - Uniform interface
  - Client-server
  - Stateless
  - Cacheable or non-cacheable
  - Layered system
  - Code-on-demand (optional)
- For applications to communicate over a network, networking protocols must be used to facilitate those communications
  - For REST APIs, HTTP(S) is the most common choice
- **For CCNA**
  - CRUD
  - HTTP client request verbs
  - HTTP server response codes
  - Basic characteristics of REST APIs

## Client-Server

- REST uses a client-server architecture
- The client uses API calls (HTTP Requests) to access the resources on the server
- The separation btw the client and server means they both change and evolve independently of each other
  - When client/server application changes, the interface btw them must not break



## Stateless

- REST APIs exchanges are stateless
- This means that each API exchange is a separate event, independent of all past exchanges btw the client and server
  - The server does not store information about previous requests from the client to determine how it should respond to new requests
- If authentication is required, client must authenticate with the server for every request
- TCP is an example of stateful protocol
- UDP is an example of stateless protocol
- Note: \*Although REST APIs use HTTP, which uses TCP (stateful) as its layer 4 protocol, HTTP and REST APIs themselves are not stateful
  - The functions of each layer (layer 4 and 7) are separate

## Cacheable / Non-cacheable

- REST APIs must support caching of data
- Caching refers to storing data for future use
- Not all resources have to be cacheable, but cacheable resources MUST be declared as cacheable

## Cisco DevNet

- Cisco application used for automation

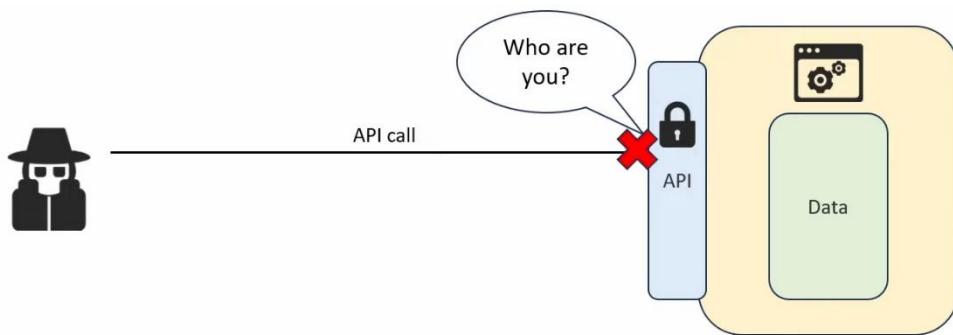
## Day 61 (2) - REST API Authentication

### Things covered

- Authentication
- Authentication types

## REST API Authentication

- Authentication is the process of validating the identity of a user or system to ensure legitimate access to resources
  - For API, the resource is the application and its data
- W/o authentication, unauthorized users can send API requests, potentially accessing sensitive data or modifying the application
  - Implementing a reliable authentication method is essential for protecting applications and data
- Many APIs track usage for analytics and billing purposes
  - E.g. charging customers according to how much they use the API



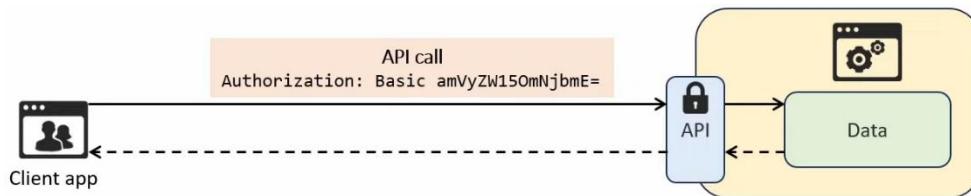
### Types of REST API authentication

- Also called 'methods' or 'schemes'
- We will cover 4 types
  - Basic authentication
    - Sends a username and password in every request, encoded in Base64
  - Bearer authentication
    - Uses a token (bearer token) as an HTTP header in each of the request to verify the client's identity
  - API key authentication
    - Requires a unique key, typically included as a HTTP header, to authenticate API requests
  - OAuth2.0
    - A secure framework that grants access via access token, commonly used for delegated access and third-party authentication

## Basic Authentication



- Includes a username and password in the HTTP headers of each API request for authentication
- Credentials are encoded in Base64 format but not encrypted
  - Better to use HTTPS (TLS) for security
- Username/password are sent in the format "username:password", encoded in Base64



- Advantage
  - Simple and easy to implement
- Disadvantage
  - Since credentials are used in every request, attackers could steal them if the connection is not properly secured
  - Even if using HTTPS for encryption, relying solely on a username/password combination isn't particularly secure

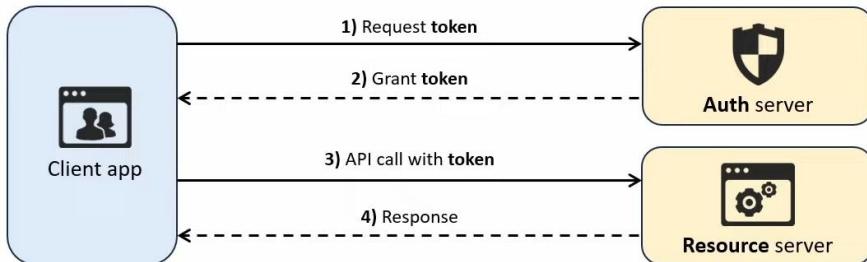
## Bearer Authentication

- Uses a token instead of username/password
  - A form of token-based authentication
- How it works
  - Client first obtain the token by authenticating with an authorization server
    - Could be done using Basic Authentication or another method
  - For each API call, the client includes the token in the HTTP Authorization header
    - E.g.

Authorization: Bearer ya29.a0ARrdaM8

- 'Bearer': anyone who possess the token can use it

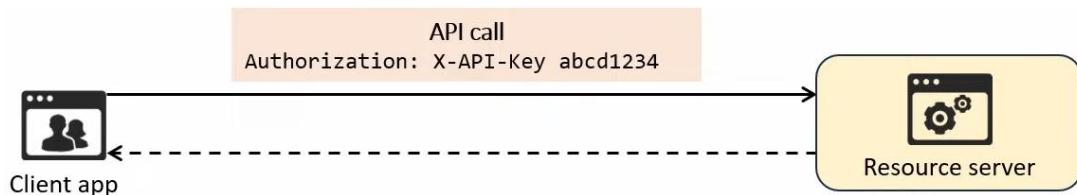
- If an attacker steals the token, they can make API calls as if they were the legitimate user
- To mitigate against this, tokens expire after a set period of time



- Advantage
  - More secure than Basic Authentication (no need to transmit the username/password for every API call)
  - Tokens expire, so a stolen token will only be temporarily valid
- Disadvantage
  - If a token is stolen, the attacker can access the API until it expires
  - Tokens need to be refreshed periodically, adding extra complexity to implement
  - Should only be used with HTTPS

## API Key Authentication

- Uses a static key issued by the API provider
  - Client uses this key in each API call for authentication
  - Key is static and remains valid until revoked
- API key can be sent in
  - HTTP Authorization header (recommended)
  - URL - not recommended since URLs are often logged by web servers, browsers, etc
  - Cookie - sometimes used for web-based APIs

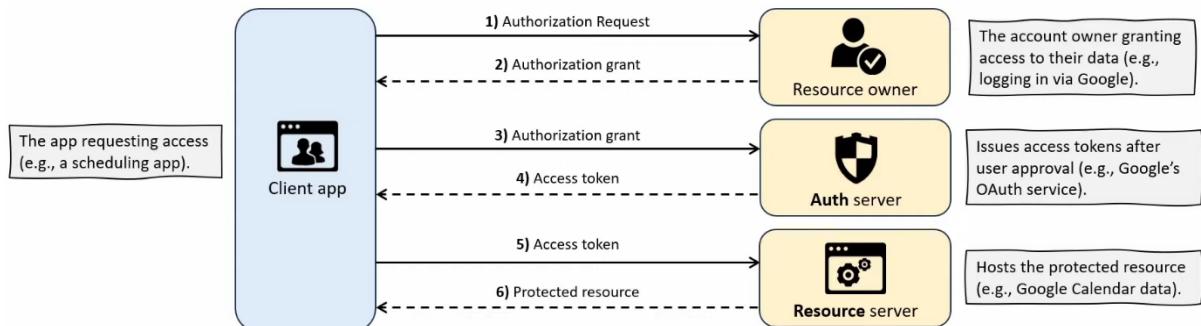


- Advantages
  - Easier to implement than Bearer authentication (no need to refresh tokens)
  - Good for tracking API usage, often used by cloud services and third-party APIs
- Disadvantage
  - If stolen, the key grants full access until revoked
  - API keys must be rotated manually to maintain security, whereas tokens expire automatically

## OAuth2.0

- A secure authentication framework that is widely used in modern web applications
- Provides access delegation, granting third-party applications limited access to resources on behalf of the resource's owner
  - No need to share the resource owner's credentials with the third party
- Examples
  - Logging in with Google
    - Many websites and apps offer the option to log in using your Google account
  - Connecting apps to social media account
    - Many apps can be connected to accounts on social media platforms like Instagram and Facebook
- The authentication/authorization process consists of 6 steps
  1. Client app requests authorization from the resource owner (you) to access the resource (your Google calendar data)
  2. Resource owner grants authorization by logging into their account (e.g. Google) and giving permission
  3. Client app exchanges the authorization grant for an access token from the auth server
  4. Auth server provides an access token to the client app
  5. Client app sends the access token to the resource server (e.g. Google's server hosting calendar data) to request the resource
  6. Resource server validates the access token and provides the requested resource (calendar data) to the client app
- The access token granted in step 4 functions just like the token used in bearer authentication
  - Grants access to the specified resource within the appropriate scope of access (e.g. read-only access)

- Access tokens expire after a short period, but Auth2.0 uses refresh tokens (granted by Auth server) to obtain new access token w/o requiring the user to login every time



## Summary

- REST API authentication ensures only authorized users or systems can access an API.
  - Without authentication, unauthorized users could access or modify data.
- Four types (methods/schemes) of REST API authentication:
  - **Basic Authentication**
    - Uses a **username and password** encoded in **Base64**, but not encrypted.
    - Requires **HTTPS (TLS)** for security since cleartext credentials are sent with each request.
  - **Bearer Authentication**
    - Uses a **bearer token**, granted by an Auth server, instead of a username/password for authentication.
    - Tokens **expire** after a short time but can be stolen if not protected.
  - **API Key Authentication**
    - Uses a **static key** issued by the API provider for authentication.
    - Good for tracking API usage.
    - Easier to implement but **less secure** since stolen keys remain valid until revoked.
  - **OAuth 2.0**
    - Provides **access delegation**, allowing third-party apps **limited access** to resources.
    - Uses **access tokens** that expire and can be refreshed (with a **refresh token**) without user reauthentication.
    - Four main parties:
      - Resource owner
      - Client app
      - Auth server
      - Resource server

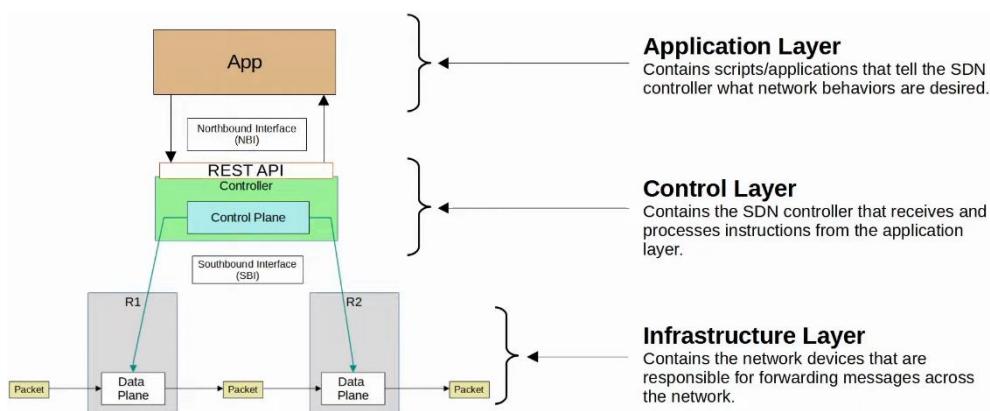
## Software Defined Networking

### Things covered

- SDN Review
- Cisco SD-Access
- Cisco DNA Center
- DNA Center network management vs traditional

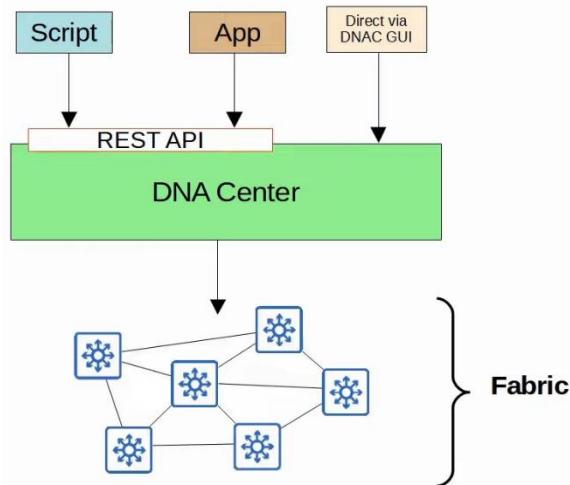
## SDN Review

- Software-defined networking (SDN) is an approach that centralizes the control plane into an application called a controller
- Traditional control planes use a distributed architecture
- Controller can interact programmatically with network devices using APIs

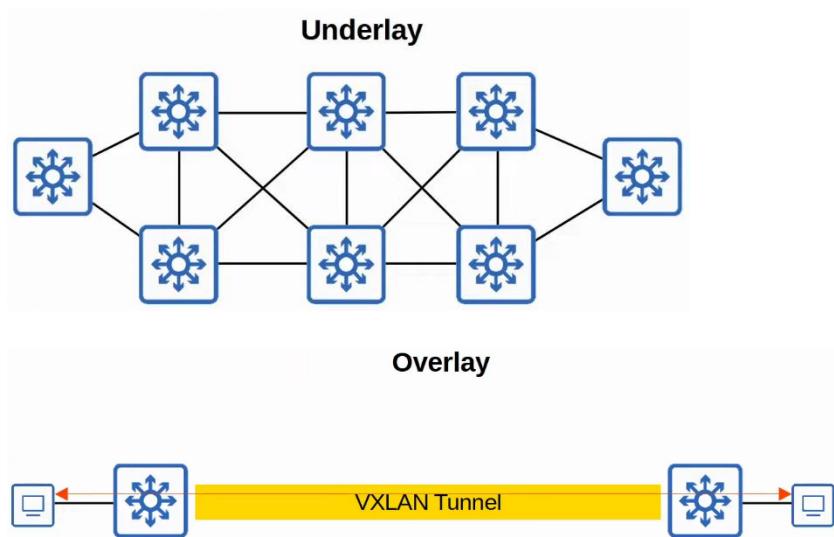


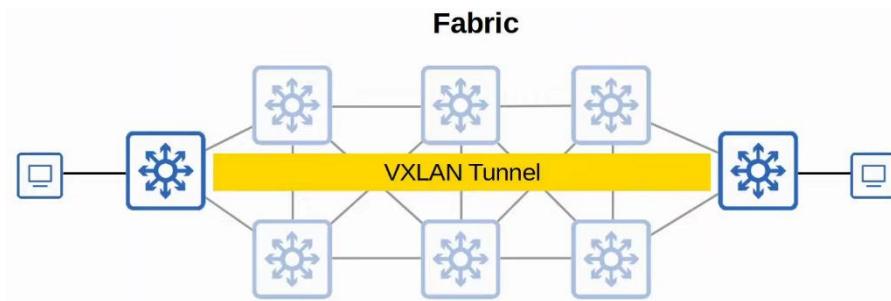
## Cisco SD-Access

- This is Cisco's SDN solution for automating campus LANs
  - ACI (Application Centric Infrastructure) is their SDN solution for automating data center networks
  - SD-WAN is their SDN solution for automating WANs
- Cisco DNA (Digital Network Architecture) Center is the controller at the center of SD-Access



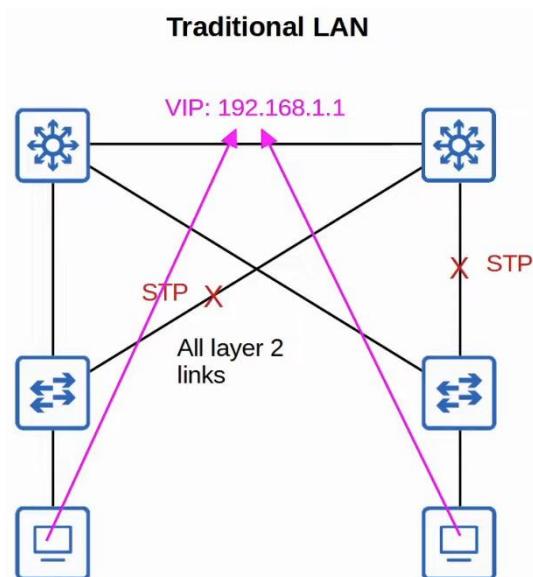
- Underlay
  - The underlying physical network of devices and connections (including wired and wireless) which provide IP connectivity (i.e. using IS-IS)
  - Multilayer switches and their connections
- Overlay
  - The virtual network built on top of the physical underlay network
  - SD-Access uses VXLAN (Virtual Extensible LAN) to build tunnels
- Fabric
  - Combination of the overlay and underlay
  - The physical and virtual network as a whole



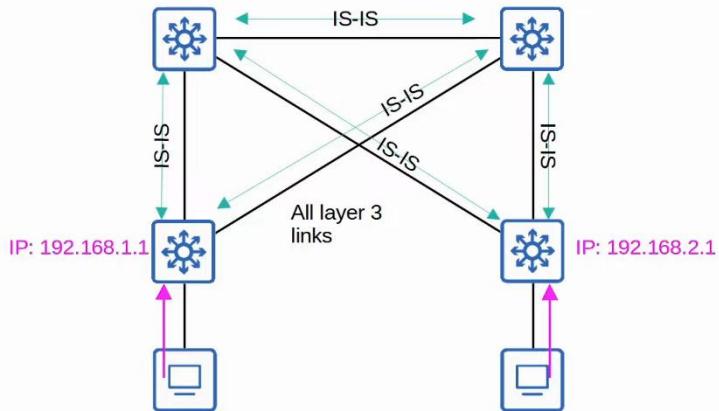


### SD-Access Underlay

- Purpose is to support the VXLAN tunnels of the overlay
- There are 3 different roles for switches in SD-Access
  - Edge nodes
    - Connect to end hosts
  - Border nodes
    - Connect to devices outside of the SD-Access domain, i.e. WAN routers
  - Control nodes
    - Use LISP (Locater ID Separation Protocol) to perform various control plane functions
- You can add SD-Access on top of an existing network (brownfield deployment) if your network hardware and software supports it
  - In this case, DNA Center won't configure the underlay
- A new deployment (greenfield deployment) will be configured by DNA center to use the optimal SD-Access underlay
  - All switches are Layer 3 and use IS-IS as their routing protocol
  - All links btw switches are routed ports. This means STP is not needed
  - Edge nodes (access switches) act as the default gateway of end hosts (routed access layer)

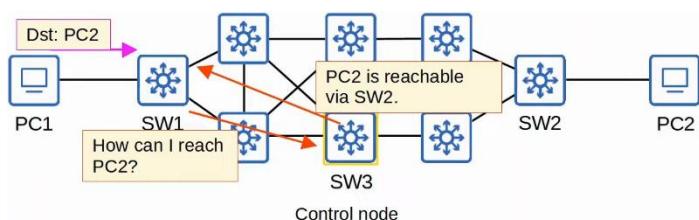
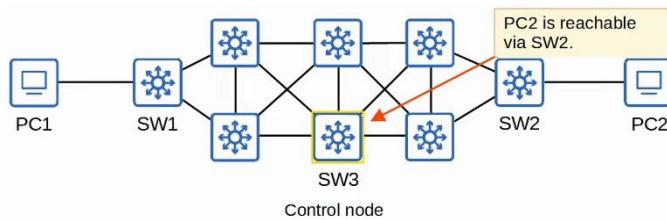


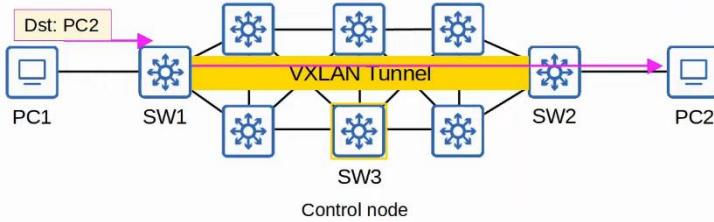
### SD-Access Underlay



### SD-Access Overlay

- LISP provides the control plane of SD-Access
  - A list of mappings of EID (endpoint identifiers) to RLOCs (routing locators) is kept
  - EIDs identify end hosts connected to edge switches, and RLOCs identify the edge switch which can be used to reach the end host
- Cisco TrustSec (CTS) provides policy control (QoS, security policy, etc)
- VXLAN provides the data plane of SD-Access





## Cisco DNA Center

- Has 2 main roles
  - SDN controller in SD-Access
  - Network manager in a traditional network (non-SD-Access)
- DNA Center is an application installed on Cisco UCS server hardware
- It has a REST API which can be used to interact with DNA Center
- The SBI supports protocols such as NETCONF and RESTCONF (as well as traditional protocols like Telnet, SSH, SNMP)
- DNA Center enables Intent-Based Networking (IBN)
  - Allow the engineer to communicate their intent or network behaviour to DNA Center, and then DNA Center will take care of the details of the actual configurations and policies on devices
- Example of IBN
  - Traditional security policies using ACLs can become very cumbersome
    - ACLs have thousands of entries
    - The intent of entries is forgotten with time and as engineers leave and new engineers take over
    - Configuring and applying ACLs correctly across a network is cumbersome and leaves room for error
  - DNA Center allows the engineer to specify the intent of the policy (e.g. this group of users can't communicate with this group, this group can access this server but not that server), and DNA Center will take care of the details of implementing the policy

## DNA Center vs Traditional Network Management

- Traditional network management
  - Devices configured 1-by-1 via SSH/console
  - Devices configured manually via console before being deployed
  - Configs and policies are managed per device (distributed)
  - New network deployments can take a long time due to manual labour required
  - Errors and failures are more likely due to increased manual effort
- DNA Center-based network management
  - Devices centrally managed and monitored from the DNA Center GUI or other applications using its REST API
  - Administrator communicates their intended network behaviour to DNA Center, which changes those intentions into configs on the managed devices
  - Configs and policies are centrally managed
  - Software versions are also centrally managed. DNA Center can monitor cloud servers for new versions and then update the managed devices
  - New network deployments are much quicker. New devices can automatically receive their configs from DNA Center w/o manual config

## Ansible, Puppet, Chef

### Things covered

- Intro to configuration management tools
- Ansible
- Puppet
- Chef

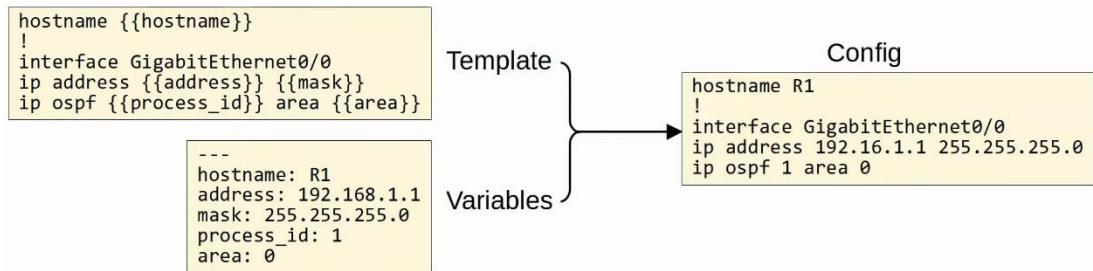
## Configuration Drift

- When individual changes made over time cause a device's config to deviate from the standard/correct configs as defined by the company
  - Most devices will have configs defined by standard templates by network architects
- Even without automation tools, best to have standard configuration management practices
  - E.g. when a change is made, save the config as text file and place it in a shared folder

## Configuration Provisioning

- Refers to how configuration changes are applied to devices
  - Includes configuring new devices

- Traditionally done 1-by-1 via SSH
  - Not practical in large networks
- Configuration management tools like Ansible, Puppet and Chef allow us to make changes to devices on a mass scale with a fraction of the time/effort
- 2 essential components
  - Templates
  - Variables



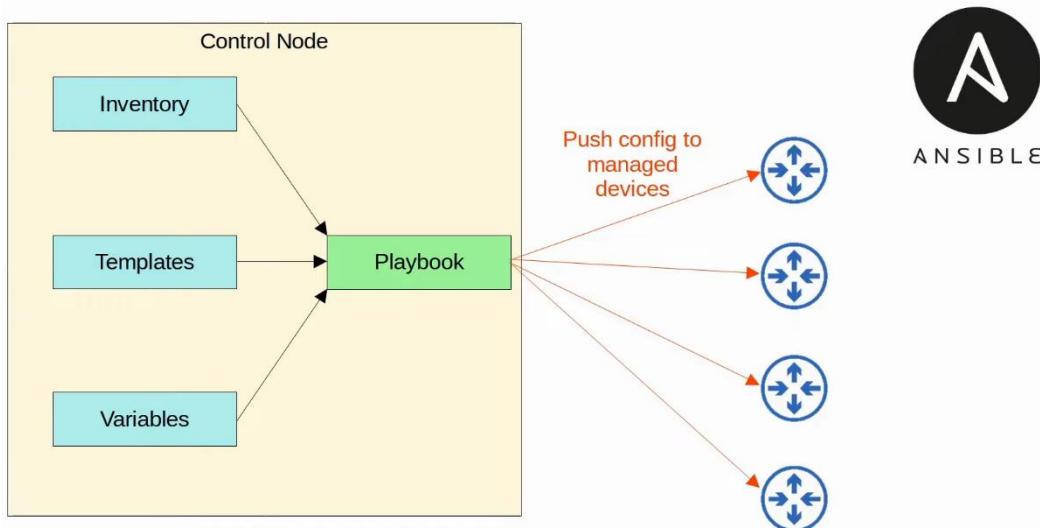
## Configuration Management Tools

- They are network automation tools that facilitate the centralized control of large numbers of network devices
- E.g. Ansible, Puppet, Chef
  - Originally developed for VMs, to enable server system admins to automate the process of creating, configuring, and removing VMs
  - Also widely used to manage network devices
- Can perform tasks such as
  - Generate configs for new devices on a large scale
  - Perform config changes on devices (all device, or some devices)
  - Check device config for compliance with defined standards
  - Compare config btw devices, and btw different versions of configs on the same device

## Ansible

- Config management tool owned by Red Hat
- Written in Python
- Agentless
  - Does not require any special software to run on managed device
- Use SSH to connect to devices, make config changes, extract information, etc
- Push model

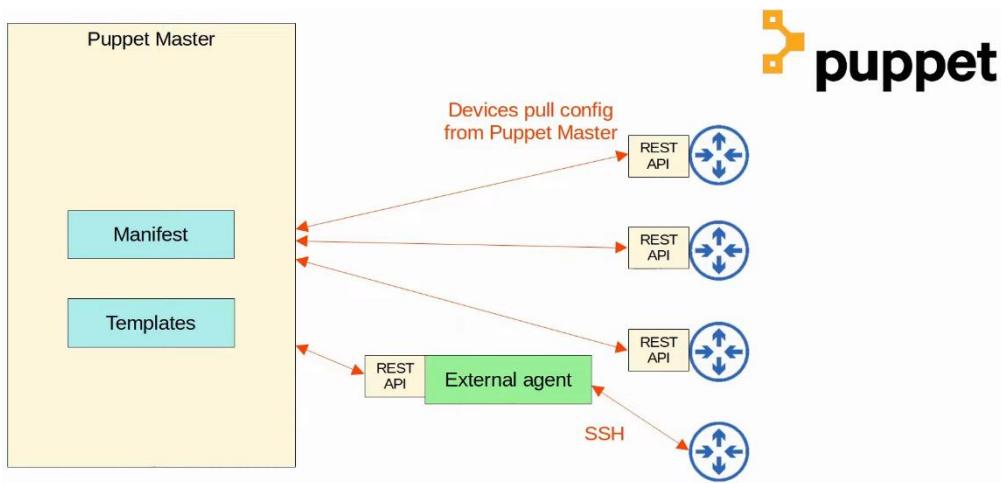
- Ansible server (Control node) uses SSH to connect to managed devices and push config changes to them
  - Puppet and Chef - Pull model
- After installing Ansible, must create several text files
  - Playbooks
    - 'Blueprints of automation tasks'
    - Outline the logic and actions of the tasks that Ansible should do
    - Written in YAML
  - Inventory
    - List the devices that will be managed by Ansible
    - List the characteristics of each device (e.g. device role - access switch, core switch, WAN router, firewall, etc)
    - Written in INI, YAML, etc
  - Templates
    - Represent a device's config file
    - Specific values for variables are not provided
    - Written in Jinja2
  - Variables
    - List the variables and their values
    - Values are substituted into the templates to create complete config files
    - Written in YAML



## Puppet

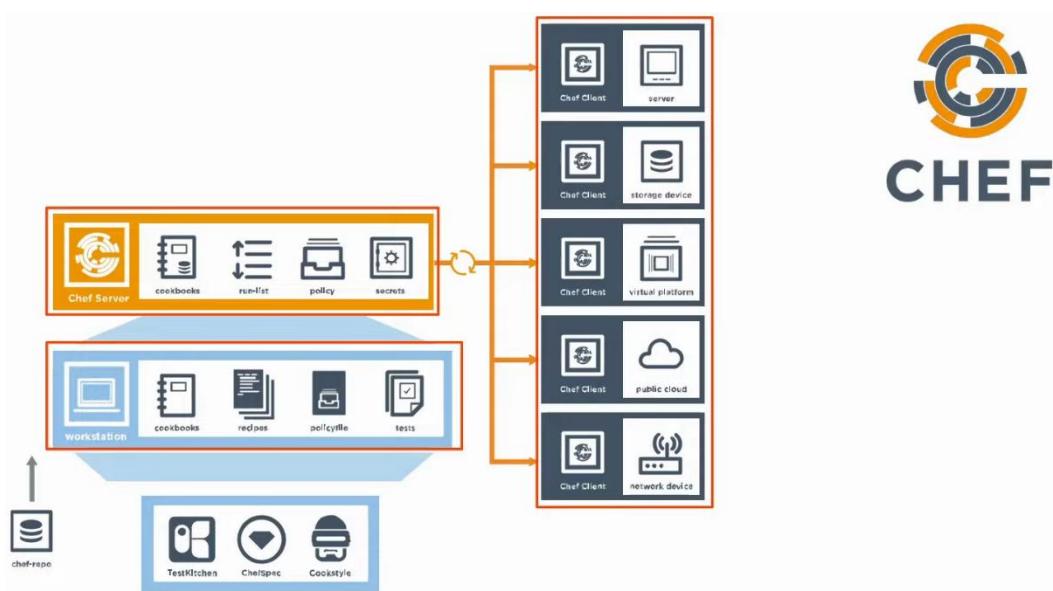
- Written in Ruby
- Agent-based
  - Specific software must be installed on the managed device
  - Not all Cisco devices support a Puppet agent
- It can be run agentless

- A proxy agent runs on an external hosts, and the proxy agent uses SSH to connect to the managed devices and communicate with them
- Puppet server called 'Puppet master'
- Pull model
  - Clients pull configuration from the Puppet master
  - Clients use TCP 8140
- Uses a proprietary language for files
- Text files required on the Puppet master
  - Manifest
    - Defines the desired config state of a network device
  - Templates
    - Similar to Ansible templates, used to generate manifests



## Chef

- Written in Ruby
- Agent-based
  - Specific software must be installed on managed devices
  - Not all Cisco device support a Chef agent
- TCP 10002
- Files use a DSL (Domain-specific Language) based on Ruby
- Text files used
  - Resources
    - 'Ingredients' in a recipe
    - Config objects managed by Chef
  - Recipes
    - 'Recipe' in a cookbook
    - Outline the logic and actions of the tasks performed on the resources
  - Cookbooks
    - A set of related recipes grouped together
  - Run-list
    - An ordered list of recipes that are run to bring a device to the desired config state



## Comparison

	<b>Ansible</b>	<b>Puppet</b>	<b>Chef</b>
Key Files defining actions	Playbook	Manifest	Recipe, Run-list
Communication Protocol	SSH	HTTPS (via REST API)	HTTPS (via REST API)
Key Port	22 (SSH port)	8140	10002
Agent-based/ Agentless	Agentless	Agent-based (or Agentless)	Agent-based
Push/Pull	Push	Pull	Pull

# Terraform

## Things covered

- Infrastructure as Code
- Provisioning & Management
- Mutable/Immutable infrastructure
- Procedural vs Declarative
- Terraform Basics

## Infrastructure as Code

- Practice of managing & provisioning infrastructure (e.g. servers, networks, clouds, etc) using machine-readable config files (code) instead of manual config (e.g. CLI/GUI)
- Ansible, Puppet, and Chef are examples of IaC configuration management tools
- Terraform
  - An IaC-based provisioning tool that automates the creation of infrastructure resources
- IaC automates infrastructure deployment and management, ensuring consistency, scalability, and repeatability

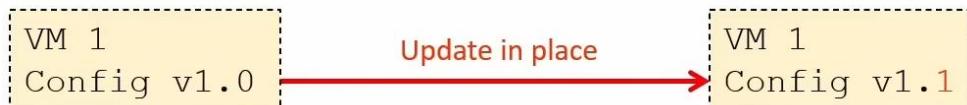
## Provisioning Vs Management

- Config management (e.g. Ansible, Puppet, Chef)
  - Manages existing infrastructure by installing software, configuring settings, and maintaining system state
  - Ensures consistency by applying and enforcing configurations across multiple devices
- Infrastructure provisioning (e.g. Terraform)
  - Creates, modifies, and deletes infrastructure resources such as servers and network infrastructures
  - Focus on initial setup rather than ongoing configuration management
- Config management tools work on already existing systems
- Provisioning tools build infrastructure from scratch
- Terraform and Ansible can work together

## Mutable vs Immutable Infrastructure

- Config management tools typically use a mutable infrastructure approach

- Infra can be modified after deployment (e.g. applying updates, patches, or config changes)
- Changes are made in place, meaning existing resources are updated rather than replaced



- Provisioning tools employ an immutable infra approach
  - Infra cannot be changed after deployment
    - "Changes" involve replacing the previous resource with a new version
  - No config drift, since each deployment starts from a fresh, predefined state



## Procedural vs Declarative Approach

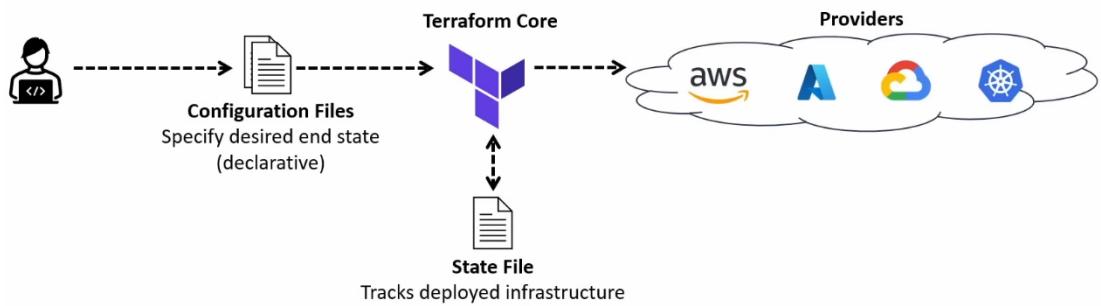
- Procedural approach (aka Imperative)
  - Follows explicit steps in a specific order to achieve the desired outcome
  - The user must define each action to config the infra
  - Provides greater control compared to a declarative approach
- Declarative approach
  - Defines the desired end state
  - The tool (e.g. Terraform) figures out the steps needed to achieve the goal
  - Easier to maintain and ensures consistency across deployments

PROCEDURAL	DECLARATIVE
Ansible	Terraform
Chef	Puppet

## Terraform

- Open source IaC tool developed by HashiCorp

- Primarily a provisioning tool, focused on deploying infra resources on various cloud & on-prem platforms
  - These platforms are called providers
  - E.g. AWS, Azure, etc
  - Includes integrations with Cisco platforms like Catalyst Center, ACI, and IOS XE
- Push model, agentless
  - Does not require a software agent on infra it provisions or manages



- The basic Terraform workflow consists of 3 main steps
  - Write
    - Define the desired state of your infra resources in config file
  - Plan
    - Verify the changes that will be executed before applying them
  - Apply
    - Execute the plan to provision and manage the infra resources
- Terraform core written in Go, config files written in HashiCorp Configuration Language (HCL)
  - HCL is a Domain Specific Language (DSL)
  - DSLs allow users to perform complex tasks with much less effort than a general language



## Summary

- **Infrastructure as Code (IaC)** is the practice of provisioning and managing infrastructure (servers, networks, cloud resources) using machine-readable configuration files (code) instead of manual configuration (e.g., CLI/GUI).
- **Configuration management** tools (e.g., Ansible, Puppet, Chef) focus on managing existing infrastructure by installing software, configuring settings (e.g., router configurations), and maintaining system state.
- **Infrastructure provisioning** tools (e.g., Terraform) focus on creating, modifying, and deleting infrastructure resources.
- **Mutable infrastructure** can be modified after deployment (e.g., applying updates, patches, or configuration changes).
- **Immutable infrastructure** cannot be changed after deployment; changes involve replacing the previous resource with a new version.
- A **procedural** (imperative) approach follows explicit steps in a specific order to achieve the desired outcome.
- A **declarative** approach defines the desired end state, and the IaC tool figures out the steps needed to achieve the goal.
- **Terraform** is an open-source IaC tool developed by HashiCorp.
  - It is primarily a provisioning tool used for deploying infrastructure on **providers** like AWS, Azure, GCP, Kubernetes, etc.
    - It interacts with these providers via their APIs.
  - Like Ansible, it uses a **push** model and is **agentless**.
  - Main components: **Terraform Core**, **configuration files**, **state file**, **providers**.
  - The basic Terraform workflow consists of three main steps:
    - **Write**: Define the desired state of your infrastructure resources in configuration files.
    - **Plan**: Verify the changes that will be executed before applying them.
    - **Apply**: Execute the plan to provision and manage the infrastructure resources.
  - Terraform Core is written in **Go**, and configuration files are written in **HashiCorp Configuration Language (HCL)**, a DSL.